MULTI-VIEW VIDEO CODING VIA DENSE DEPTH FIELD

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

BURAK OĞUZ ÖZKALAYCI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR

THE DEGREE OF MASTER OF SCIENCE

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2006

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. İsmet Erkmen
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. A. Aydın Alatan
Supervisor

Examining Committee Members

Assoc. Prof. Dr. Gözde B. Akar (METU) _____

Assoc. Prof. Dr. A. Aydın Alatan (METU) _____

Prof. Dr. Murat Aşkar (METU) _____

Asst. Prof. Dr. Çağatay Candan (METU) _____

Prof. Dr. Levent Onural (Bilkent University) _____

"I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work."

Name Surname : BURAK OĞUZ ÖZKALAYCI

Signature :

ABSTRACT

MULTI-VIEW VIDEO CODING VIA DENSE DEPTH FIELD

Özkalaycı, Burak Oğuz

M.S., Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. A. Aydın Alatan

September 2006, 104 pages.

Emerging 3-D applications and 3-D display technologies raise some transmission problems of the next-generation multimedia data. Multi-view Video Coding (MVC) is one of the challenging topics in this area, that is on its road for standardization via ISO MPEG. In this thesis, a 3-D geometry-based MVC approach is proposed and analyzed in terms of its compression performance. For this purpose, the overall study is partitioned into three preceding parts. The first step is dense depth estimation of a view from a fully calibrated multi-view set. The calibration information and smoothness assumptions are utilized for determining dense correspondences via a Markov Random Field (MRF) model, which is solved by Belief Propagation (BP) method. In the second part, the estimated dense depth maps are utilized for generating (predicting) arbitrary (other camera) views of a scene, that is known as novel view generation. A 3-D warping algorithm, which is followed by an occlusion-compatible hole-filling process, is implemented for this aim. In order to suppress the occlusion artifacts, an intermediate novel view generation method, which fuses two novel views generated from different source views, is developed. Finally, for the last part, dense depth estimation and intermediate

novel view generation tools are utilized in the proposed H.264-based MVC scheme for the removal of the spatial redundancies between different views. The performance of the proposed approach is compared against the simulcast coding and a recent MVC proposal, which is expected to be the standard recommendation for MPEG in the near future. These results show that the geometric approaches in MVC can still be utilized, especially in certain 3-D applications, in addition to conventional temporal motion compensation techniques, although the rate-distortion performances of geometry-free approaches are quite superior.

Keywords: Multi-view Video Coding, Dense Depth Estimation, Novel View Generation, Markov Random Field, Belief Propagation.

ÖZ

SIK DERİNLİK HARİTASI İLE ÇOK-GÖRÜNTÜLÜ VİDEO KODLAMASI

Özkalaycı, Burak Oğuz
Yüksek Lisans, Elektrik Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Assoc. Prof. A. Aydın Alatan

Eylül 2006, 104 sayfa.

Gelişmekte olan 3-B uygulamalar ve görüntüleme teknolojileri, beraberinde yeni nesil çoklu ortam verilerinin iletimi problemlerini gündeme getirmektedir. Bu alandaki güncel çalşmalardan biri de hızla standartlaşma yolunda ilerleyen Çok-görüntülü Video Kodlamasıdır (ÇVK). Bu tezde geometri temelli bir ÇVK yaklaşımı önerilmiş, sıkıştırma performansı açısından incelenmiştir. Bu amaçla, tüm çalışma birbiri üzerine dayalı üç temel bölüme ayrılmıştır. Bu adımlardan ilkinde, tam kalibre edilmiş çoklu-görüntü kümesinden, bir görüntüye ait sık derinlik haritasının tahminidir. Kalibrasyon bilgisi ve sahnenin derinlik açısından pürüzsüzlüğü varsayımı kullanılarak, sık eşlenik bulma problemi Markov Rasgele Alanlar (MRA) şeklinde modellenmiş ve Yargı Yayılımı (YY) metodu ile çözülmüştür. İkinci bölümde, kestirilen sık derinlik haritalarının da yardımıyla, sahnenin her hangi başka bir görüntüsü elde edilmiştir. Bu amaçla örtük alanlarla uyumlu bir boşluk doldurmanın ardından bir 3-B eğriltme algoritması uygulanmıştır. Örtük alanların neden olduğu sorunların azaltılması amacıyla iki farklı görüntüden elde edilen yeni görüntülerin kaynaştırılmasına dayalı yeni özgün bir ara görüntü bulma metodu geliştirilmiştir. Son bölümde ise, çoklu görüntüler arasındaki

uzamsal fazlalık bilginin arındırılması amacıyla sık derinlik kestirimi ve yeni ara görüntü yaratımı metotlarını kullanan H.264 temelli bir ÇVK yöntemi önerilmiştir. Önerilen yöntemin performansı, tüm görüntülerin ayrı ayrı kodlanması ve MPEG standardı olması beklenen ÇVK önerisi ile karşılaştırılmıştır. Benzetim sonuçları ÇVK'da geleneksel hareket kestirimi tekniklerinin yanında, özellikle belirli uygulamalarda, 3-B geometrik yaklaşımların da kullanılmasının faydalı olacağını göstermiştir.

Anahtar Kelimeler: Çok-görüntülü Video Kodlama, Sık Derinlik Kestirimi, Yeni Görüntü Yaratımı, Markov Rasgele Alanlar, Yargı Yayılımı.

ACKNOWLEDGMENTS

I would like to express my deep and sincere gratitude to my supervisor, Assoc. Prof. Dr. Aydın Alatan, for his guidance, stimulation, encouragement, confidence and friendship throughout the research.

It was much more fun to be a part of Multimedia Research Group (MMRG). I would like to thank everybody in the MMRG for their support and friendship.

I would like to also express my thanks to my colleagues and my friends Engin Tola and Birant Örten. I learned so much from them.

I would like to thank my dear friend Can Eroğul for his patience and efforts in helping and understanding me.

Finally, I would like to thank my family for their understanding and support; especially to my mother.

This work is supported by EC within FP6 under Grant 511568 with the acronym 3DTV

TABLE OF CONTENTS

ABSTRACT iv

ÖZ vi

ACKNOWLEDGMENTS viii

TABLE OF CONTENTS ix

LIST OF FIGURES xii

LIST OF ABBREVIATIONS xiv

1 INTRODUCTION 1

1.1 Scope of the Thesis 2

1.2 Outline of the Thesis 3

2 DENSE DEPTH ESTIMATION 5

2.1 Camera Model 5

2.1.1 Basic Pinhole Camera Model 6

2.1.1.1 Enhancing the Camera Model 7

2.2 Epipolar Geometry 9

2.2.1 The Fundamental Matrix 10

2.3 Clues for the Depth Estimation 12

2.4 Dense Depth Estimation in Literature 14

2.4.1 Matching Cost Measures 15

2.4.2 Aggregation of Matching Cost 16

2.4.3 Optimization of the Cost 17

2.5 MRF-based Dense Depth Estimation 18

2.5.1 Definition of Markov Random Fields 19
2.5.1.1 MRF - Gibbs Random Field Equivalence 20
2.5.2 A MRF Model for Dense Depth Estimation . . . 23
2.5.2.1 Color Consistency Constraint 24
2.5.2.2 Smoothness Constraint 27
2.5.3 The Solution of the MRF Model 29
2.5.3.1 Iterated Conditional Modes (ICM) . 30
2.5.3.2 Belief Propagation (BP) 32
2.6 Simulations on Dense Depth Estimation 36

3 NOVEL VIEW GENERATION 41
3.1 3D Scene Representations 41
3.2 Image-Based Rendering (IBR) 43
3.2.1 3D Warping Algorithm 46
3.3 Hole Filling 50
3.4 Intermediate Novel View Generation 56

4 MULTI-VIEW VIDEO CODING 61
4.1 Video Coding Fundamentals and H.264 Standard 62
4.1.1 Major Features of H.264 63
4.1.1.1 Intra Coding 64
4.1.1.2 Inter Coding 66
4.1.1.3 Transform Coding 67
4.1.1.4 Entropy Coding 68
4.2 Multi-view Video Coding 69
4.3 The Proposed MVC Method 75
4.4 Simulations of the Proposed MVC Method 80
4.4.1 Implementation of Proposed MVC Method . . . 81
4.4.2 Simulation Results 82

5 SUMMARY AND CONCLUSIONS 87
5.1 Future Work 90

APPENDICES 92

A OCCLUSION-COMPATIBLE ORDERING 92

B CALIBRATION DATA FOR BREAKDANCER AND BALLET SEQUENCES 96
B.1 Breakdancer Sequence 96
B.2 Ballet Sequence 97

REFERENCES 99

LIST OF FIGURES

2.1 Basic pin-hole camera model 6
2.2 Basic pin-hole camera relations 8
2.3 Epipolar geometry 9
2.4 Another interpretation of an epipolar line 10
2.5 The positive and negative epipoles 11
2.6 4- and 8-neighborhood 20
2.7 4- and 8- neighborhood cliques 21
2.8 Correspondences of a pixel for a depth value 26
2.9 The parallel depth planes sampling the 3D space 27
2.10 Smoothing clique functions 28
2.11 Graphical models for BP 33
2.12 ICM and BP comparison for stereo case 38
2.13 Dense depth estimation for multi-view 39
2.14 The effect of the smoothness constraint on the estimates 40

3.1 Generation of a layered depth image 45
3.2 A naive 3D warping result 47
3.3 Illustration of the regular mesh on the image plane 48
3.4 Visibility artifacts of an over-connected surface 49
3.5 A novel view generated by 3D warping algorithm 50
3.6 Epipolar constraint on the holes 52
3.7 The occlusion compatible scanning directions 53
3.8 Illustration of the possible sheets 54
3.9 Hole filling artifacts in concave cases 55
3.10 Hole filling result 56
3.11 Novel view generation result for *Breakdancer* 59

3.12 Novel view generation result for *Ballet* 60

4.1 The encoder structure of H.264 64
4.2 Intra estimation modes 65
4.3 Integer transforms in H.264. 68
4.4 Hierarchical B-frame based MVC 70
4.5 Depth and disparity maps of *Breakdancer* 71
4.6 View interpolation utilizing MVC 73
4.7 Disparity estimation for view interpolation 74
4.8 Reference camera selection 76
4.9 Illustration of the proposed MVC method 80
4.10 Comparison of MVC methods for *Breakdancer* 84
4.11 Portions of the bits for *Breakdancer* 85
4.12 Comparison of MVC methods for *Ballet* 85
4.13 Portions of the bits for *Ballet* 86
4.14 PSNR difference between estimates and reference views 86

A.1 Sphere representation of the cameras 93
A.2 An occlusion illustration on an epipolar plane 94
A.3 Occlusion-compatible scanning directions 95

LIST OF ABBREVIATIONS

B	Bi-predictive
BP	Belief Propagation
CABAC	Context-based Adaptive Binary Arithmetic Coding
CAVLC	Context-based Adaptive Variable Length Coding
DCT	Discrete Cosine Transform
DP	Dynamic Programming
DVD	Digital Video Disk
FVV	Free Viewpoint Video
GC	Graph Cut
GRF	Gibbs Random Field
HDTV	High Definition TV
I	Intra
JVT	Joint Video Group
LDI	Layered Depth Image
MAD	Mean Absolute Difference
MAP	Maximum A Posterior
MPEG	Motion Picture Experts Group
MRF	Markov Random Field
MSE	Mean Square Error
MVC	Multi-view Video Coding
NCC	Normalized Cross Correlation
P	Predictive
PDE	Partial Differential Equation
QP	Quantization Parameter
SA	Simulated Annealing
SI	Switch Intra
SP	Switch Predictive
VCD	Video Compact Disk
WTA	Winner Takes All

CHAPTER 1

INTRODUCTION

The 3D display technologies progressed drastically in the recent years. The glass-free auto-stereoscopic displays, which create the feeling of 3rd dimension by driving multiple views, are expected to spread into the consumer market in the very near future. Also it is sure that the emerging applications like 3D video and free viewpoint video are the predecessors of the next generation multimedia. However, besides the displaying problem, there exist many other open questions on the acquisition, representation and transmission of the 3-dimensional content, that are closely related to each other.

The multi-view video is a collection of traditional videos, capturing the same scene in a synchronized fashion. The multi-view video based 3D systems dominate the current approaches since it is compatible with the present multimedia networks and 3D display technologies. The developed tools in computer vision and computer graphics also favor and widen the opportunities of multi-view video

based approaches. However the transmission of the multi-view video data by the conventional video compression methods needs multiples of bandwidths which are not possible for most cases.

The conventional video compression methods utilize mainly the temporal redundancies in the frame sequence. In addition to the temporal redundancies, multi-view video also contains spatial redundancies between the views. The ongoing research activities on Multi-view Video Coding (MVC) are focused on the utilization of these spatial redundancies. Since the spatial redundancies between the views constitute the clues of the scene geometry in perspective of computer vision, MVC may means much more than a compression.

1.1 Scope of the Thesis

The research behind this dissertation is motivated by two main propositions. The first one is that the next generation multimedia standards will contain the 3D structure of the scene in some sense. And the second one is that the spatial redundancies between the views of multi-view video are encoded in the 3D structure already. This thesis is devoted to the investigation of the utilization of the 3D structure of the scene in MVC schemes.

The dissertation can be partitioned into three main steps in a progressive way to the proposition of a MVC approach. The first step is the estimation of the scene geometry. The scene geometry is represented as the dense depth maps of

some reference cameras. Dense disparity estimation techniques for stereo case are generalized to multi-view case and Markov random field modeling is used in particular. The next step is the generation of arbitrary views using the dense depth estimate and the texture of the source camera. A 3D warping algorithm is used for rendering novel views. Hole filling and view fusion methods are used to tackle the occlusion artifacts and to improve the quality of the novel views. In the last step novel view generation method is used to feed the H.264 based video coding scheme with the estimates of the frames to be encoded.

1.2 Outline of the Thesis

The structure of the thesis follows the aforementioned three main steps in a progressive way.

Chapter 2 is devoted to dense depth estimation from a full-calibrated multiple image set. Some background information on camera geometry and epipolar geometry is mentioned at first. A literature survey on dense depth estimation is given in advance to utilized Markov random field based approach is introduced. Simulation results of the dense depth estimation are given also.

Chapter 3 focuses on the novel view generation problem by the help of the estimated dense depth estimation method. The 3D scene representation problem is discussed and an image based representation/rendering method is introduced to generate an intermediate novel view. The occlusion artifacts and hole filling

methods are also mentioned.

Chapter 4 proposes a MVC scheme which utilizes the spatial redundancies between the views by exploiting the dense depth maps and the intermediate novel view generation tool. Since the simulations of the proposed approach are done on a test bed based on H.264 video coding standard, H.264 standard is also mentioned briefly.

Finally, Chapter 5 gives a summary of the thesis and concluding remarks of the proposed MVC approach. Some recommendations for MVC are also given for future work.

CHAPTER 2

DENSE DEPTH ESTIMATION

In this chapter, a variety of approaches for obtaining the depth (i.e. 3-D distance from the camera) of a scene through stereo or multiple views will be discussed. The chapter will start with some brief background information on the camera model and epipolar geometry. By the help of this background information, the dense depth estimation problem will be formulated. For the estimation of the dense depth parameters, a Markov Random Field (MRF) model will be used. The solution of this MRF model will be approached via different methods.

2.1 Camera Model

The information and the clues about the scene structure are observed through a camera device which can be modeled as a transformation from 3D world co-ordinate to 2D image plane coordinates. Although, the transformation between

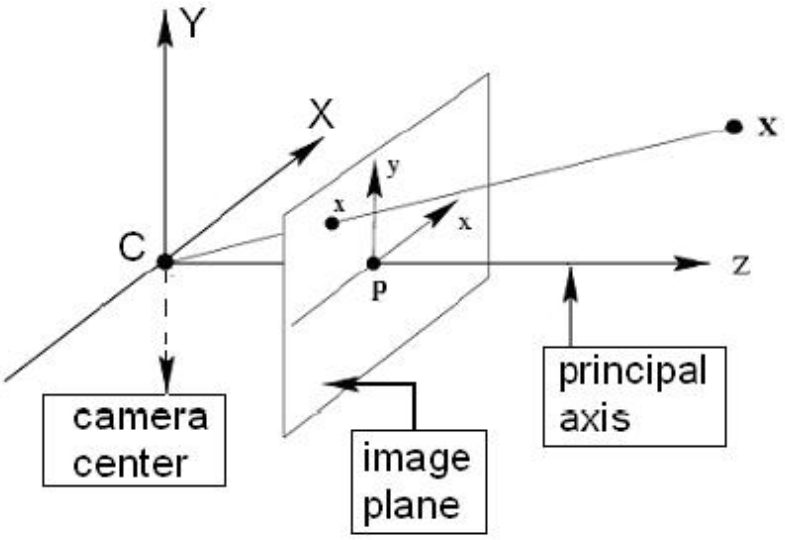

Figure 2.1: Basic pin-hole camera model

the 3D world and the 2D image plane is not linear due to the physical phenomenon, called *radial lens distortion* [1], the effects of radial lens distortion can be removed by pre-processing (or can be neglected) in most cases. The resulting reduced linear transformation is built upon a basic pinhole camera model.

2.1.1 Basic Pinhole Camera Model

Basic pinhole camera model relates the 3D world coordinates to 2D image plane coordinates by a line intersection with a plane. The line is defined by the 3D point to be transformed and a fixed point C, called *camera center* and the plane is defined as *image plane*, whose distance to the camera center is f, denoted as *focal length* (see Figure 2.1).

By utilizing the similarity of triangles, the relation between the 3D point X and the 2D point x in Figure 2.2 can be easily derived as

$$(X, Y, Z) \rightarrow (f\frac{X}{Z}, f\frac{Y}{Z}) \tag{2.1}$$

In homogenous coordinate system, such a transformation can be represented by a 3x4 matrix, which is called as *camera projection matrix* [1].

$$\begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.2}$$

Some other important definitions are also illustrated in Figures 2.1 and 2.2. The principal axis is the ray, which passes through the camera center and intersects the image plane perpendicularly. The principal point is the point which the principal axis intersects the image plane.

2.1.1.1 Enhancing the Camera Model

The camera projection matrix, which is derived by the basic pinhole camera model, does not work for many real cases due to some manufacturing imperfections. Moreover, 2.2 makes transformations from 3D world coordinate system with respect to the camera center to the image plane coordinate system, which is different from the pixelwise image coordinate system. Hence, the basic pinhole

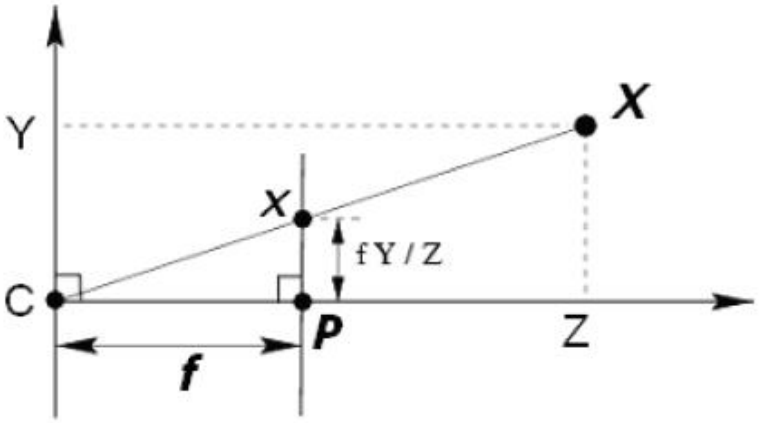

Figure 2.2: Basic pin-hole camera relations

camera model can be improved after applying an affine and an Euclidean transformations simultaneously in order to consider all the artifacts aforementioned. The improved camera projection matrix becomes

$$x = PX = K[R|t]X \tag{2.3}$$

where K is the pinhole camera model's projection matrix with an affine transformation. R and t matrices are the rotation and translation matrices of the Euclidean transformation respectively. The K matrix is called the *intrinsic parameters*, while Euclidean transformation part is denoted as the *exterior parameters* of a camera. The detailed descriptions of the affine and Euclidean transformations used for the camera model improvement are examined in [1]. A camera whose intrinsic and exterior parameters are known is denoted as *calibrated*.

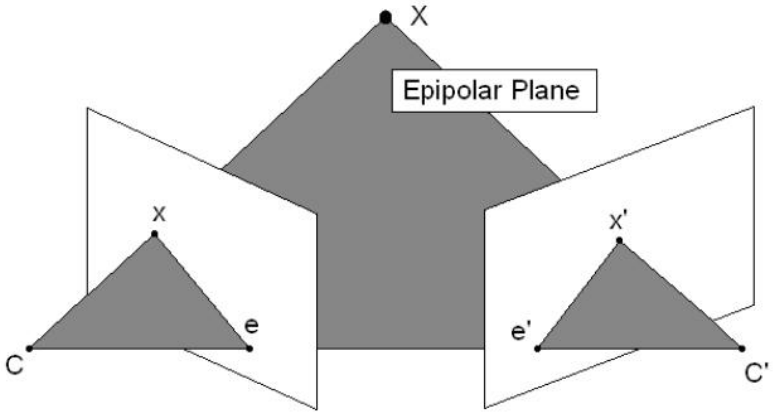

Figure 2.3: Epipolar geometry

2.2 Epipolar Geometry

The special geometric properties defined by two arbitrarily positioned cameras, which are shown in Figure 2.3, define the epipolar geometry.

The plane defined by two camera centers and an arbitrary 3D point is called the *epipolar plane* [1]. All epipolar planes pass through the line connecting the two camera centers, called the *baseline*. The intersections of the baseline with the image planes are defined as the *epipoles*. The epipoles can also be defined as the projection of a camera center to the image plane of the other camera.

The intersections of the image planes and the epipolar plane are called as *epipolar lines*. The epipolar lines corresponding to different epipolar planes always intersect at the epipoles. The Figure 2.4 shows another interpretation of the epipolar line, which is derived by the projection of the line passing from the camera center and a point, x, to the image plane of the other camera. By this

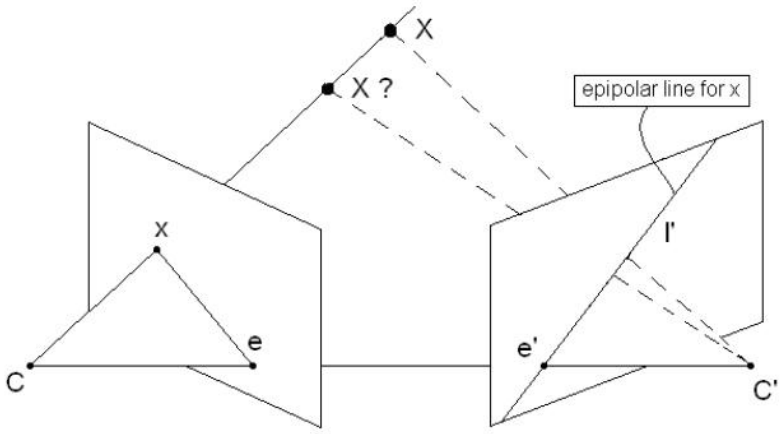

Figure 2.4: Another interpretation of an epipolar line

perspective, for every point on the image plane, there exists a corresponding epipolar line on the other image plane.

The epipoles are called as *positive* and *negative* epipoles according to positions of the camera centers and image plane on which the epipole is defined. The sign of an epipole is assigned by dot products of the vector in the direction of the principal axis of the corresponding camera and the vector connecting the camera centers. Two cases are illustrated in Figure 2.5 with the vectors mentioned above.

2.2.1 The Fundamental Matrix

The correspondence between a point and an epipolar line can be handled as a linear mapping by using the homogenous coordinate system. The 3x3 matrix denoting this mapping is called the *fundamental matrix* [1].

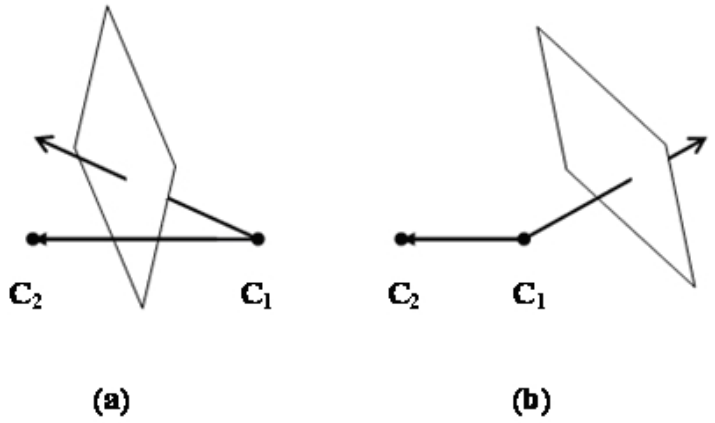

Figure 2.5: The positive and negative epipole cases are shown in (a) and (b) respectively.

The derivation of the fundamental matrix can be obtained by simply projecting the line defined by a back-projection ray, as mentioned above. The parametric equation of the back-projection ray passing though a point, x, on the image plane is given as

$$X(\lambda) = P^{+}x + \lambda C \tag{2.4}$$

where C is the camera center and P^{+} is the pseudo inverse of the projection matrix P, satisfying $PP^{+} = I$. The camera center, C, is the null space of $P(PC = 0)$, which means in homogenous coordinate system the projection of camera center is not defined.

The projections of the 3D points lying on the back-projection ray are on the corresponding epipolar line. Hence, the parametric equation of the epipolar line is obtained as

$$x'(\lambda) = P'P^{+}x + \lambda P'C = P'P^{+}x + \lambda e' \tag{2.5}$$

where e' is the epipole point. Since epipole must be on the epipolar line, the cross product of the epipole and the $x'(\lambda)$ gives the line equation of the epipolar line.

$$l' = [e']_\times P' P^+ x + \lambda [e']_\times e' = [e']_\times P' P^+ x \tag{2.6}$$

Finally the fundamental matrix, mapping a point to its corresponding epipolar line can be obtained as

$$F = [e']_\times P' P^+ \tag{2.7}$$

2.3 Clues for the Depth Estimation

During the image formation phase, the depth information of the scene points is inevitably lost due to the encountered projection. Theoretically, by using two differently positioned cameras the depth information of a scene can be revealed by using the calibration data of the cameras. After intersecting the back-projection rays of the image points corresponding to the same scene point, the 3D coordinates can be obtained by a procedure called as triangulation [1]. However, to be able to use the *triangulation* tool, the image point correspondences should be known.

For many camera calibration applications, the image point correspondences are determined sparsely, and hence, robustly by restricting the correspondence finding problem to domain of salient features of the images [1]. However, for dense depth estimation, a correspondence has to be obtained at every image

point, including the points on a constant intensity region or a repeating pattern. In order to be able to determine correct correspondences at all these points, the following two main clues are utilized widely for obtaining a solution:

- Affinity in the color or intensity
- Smooth depth variations

Expecting a similarity in color (or intensity) for an image point correspondence is rational. However, due to some scene properties, such as variations in illumination (and reflection) or some imperfections in the capturing device, such as noise or white balance differences, generally the colors of the correct correspondences do not match exactly. However, the L'ambertian [2] assumption is still a powerful tool in most cases. In order to measure the color similarity, some basic approaches, e.g. mean square error (MSE) and mean absolute difference (MAD), or some more elegant methods, which will be mentioned in the next section, can be used. However, increasing the complexity of such a measure makes it impractical due to computational complexity for the dense case. Hence, there is a tradeoff between the discriminational power as and computational cost during the selection of measure for color similarity.

The assumption that the depth map of a scene is smoothly varying, is based on the fact that the objects in most of the scenes constitute connected surfaces. In many cases, the projections of the connected surfaces inherit the connectivity

on the image plane, that is regarded as the smoothness of the depth map. Hence, the depth map of a scene can be modeled as a piece-wise continuous function in which the discontinuities belong to the object boundaries. However, the cost of imposing a smoothness constraint might cause over-smoothing the solution across the object boundaries.

2.4 Dense Depth Estimation in Literature

The dense depth estimation problem is one of the most examined topics in computer vision for the last few decades and it continues to be heavily studied. The first motivation of the problem has been acquiring the depth sense by a human-vision-like-system.

The first and the most studied case is the two parallel oriented cameras settled on a line, similar to the two eyes of a human being, that is called the *two-view stereo correspondence problem*. As the correspondences are just horizontal shifts in the left and right views, the dense depth estimation problem is generally denoted as *dense disparity estimation*. This is the reason why the disparity is often regarded as the inverse depth in computer vision.

The difficulty of the two-view stereo problem compelled the researchers to increase the number of cameras. At first, the orientations of the cameras kept similar to the two-view case, that is termed as the *multi base-line stereo problem*. As the camera calibration techniques get matured, the dense depth estimation

problem liberated to arbitrarily oriented multi camera setups.

An excellent taxonomy on two-view stereo correspondence problem is provided by Szeliski et.al. in [3]. Since a large number of the approaches for dense depth estimation is a generalization or an adaptation of the two-view stereo problem, the categorization of the dense depth estimation techniques in the literature will be achieved by following the one at [3].

The dense depth estimation problem can be partitioned into three main phases, which are calculation of the matching cost, aggregation of the matching cost (imposing a smoothness constrain) and finally the optimization of the cost. Although all the parts of the problem are closely connected to each other, the dense depth estimation methods will be mentioned according to this partition.

2.4.1 Matching Cost Measures

The most popularly used matching cost functions are aforementioned MSE and MAD due to the practical reasons in computation. However, there are also some other approaches that implicitly bind the smoothness constraint to the matching cost calculation. A typical example is is the *normalized cross correlation* (NCC) on the whole image [4]. As NCC is calculated on a support window, the spatial correlation is used for the smoothness constraint. Since the rotational effects are troublesome, some rectification tool [5],[6] should also be used to suppress them [7]. The computational cost of NCC for the whole image is also considerably

high.

Another such approach is utilizing the Fourier transform for matching the cost function [8]. Since a shift in time domain can be observed as a phase shift in frequency domain, the Fourier transform of image blocks can be used for matching. However, the occluded regions create difficulties during this approach. For matching the cost measures, the method by Birchfield [9] is shown to be insensitive to sampling artifacts.

2.4.2 Aggregation of Matching Cost

The smoothness constraint on the solution of the dense depth map is implicitly (or explicitly) defined in the aggregation of the matching costs. The aggregation methods can be grouped in two titles, as local and global approaches.

The local approaches define some kernel for smoothness and in that kernel the smoothness is explicitly imposed. In [10], a window is used as a kernel. By the assumption that in a predefined size of such a window, the depth is constant, the costs are summed up within this window. Obviously, the smoothness of the solution depends on the window size. Since the discontinuities across the object boundaries can not be preserved with a fixed size and shape window, an adaptive window method is also proposed [10],[11]. Recently, the arbitrarily shaped segments of a color clustering segmentation are also used for aggregation unit in [12], that is shown to be quite successful, while preserving the depth

discontinuities.

The global approaches impose the smoothness implicitly by modeling some relationship on the depth map. One of the popular approaches is *Markov random Field* (MRF) modeling. In MRF, the pixels of the depth field assumed to have a Markovian property in some neighborhood, that controls smoothening. The details of these methods will be given in the rest of this chapter. Apart from MRF, there are other global aggregation approaches, which define some diffusion scheme on the solution and make the smoothness diffuse on the depth map solution [13], [14]. Although the global approaches impose the smoothness in a more realistic manner, the solution of the model is cumbersome and usually only it is possible after making an approximation.

2.4.3 Optimization of the Cost

The optimization part of the problem is dependent on the approaches that are used in matching cost and cost aggregation. The most basic method for the optimization is the *winner takes all* (WTA) method [3]. The depth values, which make the aggregated costs minimum, are assigned as the solution. WTA is suitable for the locally aggregated costs. Another approach utilizes the *Dynamic Programming* (DP) on the epipolar lines [15]. Some other constraints, such as ordering or one-to-oneness, are also imposed to DP solution in [16].

For globally aggregated costs, the optimization plays a vital role on the dense

depth estimation. For MRF models, there are many proposed optimization techniques in the literature, some of which can be listed as, iterated conditional modes (ICM) [17], simulated annealing [17], Gibbs sampler [18], graph cut [19], belief propagation [20]. Diffusion-based optimization techniques are also proposed for optimizing global aggregated costs. Stochastic diffusion [21], partial differential equations (PDE) based methods [14], level set solutions [22] are typical examples for such solutions.

2.5 MRF-based Dense Depth Estimation

The spatial context of an image contains valuable information for the interpretation of an image. Unless the image is a random noise, the intensity value of a pixel is highly statistically dependent on the surrounding pixels. Markov random field is just one of the mathematical models which is used to exploit the spatial dependencies in image processing and computer vision.

In 1984, Geman and Geman introduced the MRF model for image denoising by their brilliant paper [23]. After this landmark work, MRF models have been used successfully in many other image processing and computer vision areas, such as texture modeling and classification [24], image restoration [23], image segmentation [25] and dense depth estimation [26]. In the following section, the mathematical fundamentals of MRF will be introduced.

2.5.1 Definition of Markov Random Fields

Let $X = (X_1, , X_n)$ be a family of random variables where X_i takes a value in set D. Then the family X is called as a random field and the index set $I = 1, , n$ denotes the sites of the random field. The joint event $(X_1 = x_1, , X_n = x_n)$ is simply denoted as x and it is a configuration of X, corresponding to a realization of the field. The probability of the realization of configuration x on the field X is denoted as $P(x)$.

A neighborhood system is a collection of subsets of the index set I, $\{\partial i : i \in I\}$, if the sites associated with the neighborhood ∂i satisfy the following two conditions [17]:

1. $i \notin \partial i$

2. $i \in \partial j$ if and only if $j \in \partial i$

Although, theoretically it is not obligatory, the neighborhood system is usually defined by the help of the sites surrounding the current one. The neighborhood systems that are widely used in image processing, are 4- and 8-neighborhood systems which are illustrated in Figure 2.6.

A random field X is a Markov random field, if the probability of all possible configurations of X are strictly positive and for a neighborhood system ∂, the following statement holds for every $x \in X$ [17]

Figure 2.6: 4- and 8-neighborhood

$$P(X_i = x_i | X_{I\setminus i} = x_{I\setminus i}) = P(X_i = x_i | X_{\partial i} = x_{\partial i}) \tag{2.8}$$

According to this definition, the probability of a site, given the rest of the field, is reduced to a function of neighboring sites by MRF modeling. The MRF models utilized in image processing and computer vision are generally homogenous, which means for all sites, $P(X_i = x_i | X_{\partial i} = x_{\partial i})$ have the same distribution.

2.5.1.1 MRF - Gibbs Random Field Equivalence

Gibbs Random Field (GRF) is another random field which is extensively used in statistical physics. A random field is said to be a GRF with respect to some neighborhood, ∂, if and only if its configurations obey a Gibbs distribution. A Gibbs distribution is defined as [17]

$$P(X = x) = \frac{e^{-\frac{U(x)}{T}}}{Z} \tag{2.9}$$

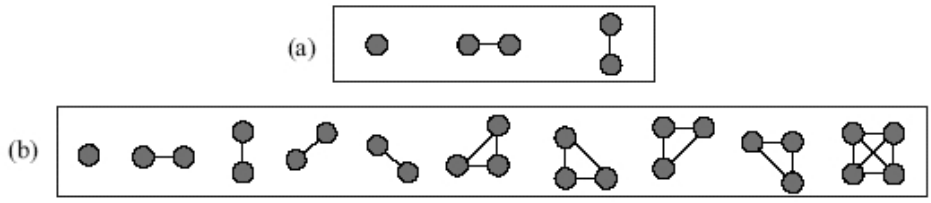

Figure 2.7: 4- and 8- neighborhood cliques

where T is a constant, denoted as *temperature*, Z is the normalizing constant, which can be obtained as

$$Z = \sum_{x \in X} e^{-\frac{U(x)}{T}} \tag{2.10}$$

and $U(x)$ is the *Gibbs energy term* which is the core point of the MRF and GRF equivalence. According to the equation 2.9 it can be concluded that, the configurations of X which have less Gibbs energy, are more probable to observe.

The clique notion, which is defined for the Gibbs energy calculation, establishes the relation between the MRF and GRF. A clique is a set of sites, in which two different indices are neighbors according to a neighborhood ∂, defined for the GRF and the set of all cliques for the field X is denoted as C_X. In Figure 2.7, the cliques defined for the 4- and 8-neighborhoods are presented.

A clique potential, V_C, of a clique is a scalar function depending on the configuration of the sites in that particular clique. For a clique containing the indices $i, j, k \in I$, the clique potential is defined as a scalar function $V_C(x_i, x_j, x_k)$. The Gibbs energy of a GRF is defined as the sum of all clique potentials defined on

the field [17].

$$U(X) = \sum_{c \in C_x} V_c(c) \tag{2.11}$$

If the clique potential functions are determined by the type of the clique formations in Figure 2.7 and invariant to position of the clique on the field then the corresponding GRF model is said to be homogenous [17].

For a random field, the GRF model imposes Gibbs distribution property in global sense, whereas the MRF model imposes a Markovian property in local sense. The Hammersley-Clifford theorem [27] combines these global and local properties by stating that a random field X, is a MRF with respect to the neighborhood system ∂ if and only if X is a GRF with respect to the same neighborhood system ∂. The proof of the theorem can be found in [28].

Beyond the theoretical assertion, the practical use of the theorem arises in modeling the behavior of the random field. By choosing appropriate clique potential functions, one can model the Markovian conditioning easily. Then, the maximum a posterior (MAP) (i.e. the most probable solution with respect to this model) can be determined by minimization of the Gibbs energy function.

In order to impose the desired constraints on the problem, defining the clique potential functions is one of the major topics in MRF modeling. Another difficult problem in MRF modeling is obtaining the configuration which minimizes the Gibbs energy. These two major topics are mentioned throughout the following

sections with some focus on the dense depth estimation problem.

2.5.2 A MRF Model for Dense Depth Estimation

The dense depth estimation problem is an ill-posed problem which should be made well-posed by imposing some constraints. The most widely used approach for converting an ill-posed problem into a well-posed one is via the *regularization* techniques [29]. For the regularization of the dense depth estimation, the smoothness assumption plays the main role. The neighborhood definition in MRF modeling provides the desired regularization affect for the solution.

The global sense of formulating the dense depth estimation problem is the second main aspect of the MRF modeling. The Markovian property on the depth field makes a direct or indirect interaction between each node in the field. Moreover, these implicit relations can be packed into the joint probability density function of the random field by exploiting the equivalence between GRF and MRF.

The explanation of the MRF model for the dense depth estimation is much more comprehensible in terms of the Gibbs energy definition. The clique energies composing the Gibbs energy are grouped into two main terms, as given in 2.12, where X represents the depth field. The first term, which is calculated from first order cliques, models the color consistency of the solution. Whereas the second term, which is calculated from the second order cliques, models the smoothness

of the solution. These kind of MRFs with a maximum clique order of 2 is denoted as *pairwise MRFs* [17]. In the next sections these terms are explained in detail.

$$U(X) = \sum_{x \in C_0} V_{C_0}(x) + \lambda \cdot \sum_{x_0, x_1 \in C_1} V_{C_1}(x_0, x_1) \tag{2.12}$$

2.5.2.1 Color Consistency Constraint

As mentioned before, the depth of a 3-D scene point can be obtained by triangulation of its correspondent pixels on two images. In reverse order, assigning a depth value to a pixel defines the correspondent pixels on the other images. For the correct depth assignment the color of all the correspondences would be expected to be similar under the same lightning conditions, following the L'ambertian assumption (i.e. a L'ambertian surface reflects equal amount of light towards all directions or cameras).

The first order clique energies, which are just a function of the configuration of a pixel in the field, are defined as the color consistency cost of the assigned depth. The color consistency function can be chosen as the sum of absolute differences of the correspondences on different views, as below

$$V_{C_0}(x) = \frac{1}{N-1} \sum_{i=1}^{N-1} |I_0(x) - I_i(f_i(d_x))| \tag{2.13}$$

The I functions represent the intensity or the RGB color maps of the views. The f_i function in 2.13 is a mapping between the assigned depth value, d_x, of a pixel, x, on the 0^{th} (reference) image and the correspondent pixel position on the i^{th}

image. The f_i function can be easily derived from the projection matrices of the 0^{th} and the i^{th} views as the back- and re-projections (see 2.4 and 2.3 respectively).

However, this kind of color consistency cost does not take the visibility constraints into account and assumes that all the pixels on the reference camera are visible by the rest of the cameras. In general, this is not the case and many occluded regions might exist on the other camera views. In order to handle such occlusions, many methods, which vary in complexity from taking the best half of the correspondences [3] to computing the occluded pixels for each depth assignment [30], have been proposed.

In Figure 2.8, the correspondences of a pixel for an assigned depth value is illustrated. For the n-image case, a back-projection is followed by n-1 projections to find the correspondences. Repeating this procedure for all pixels at all depth assignments is computationally cumbersome. For the stereo case, when two images exist, the correspondences can be searched much more easily by the rectification algorithms [5], [6] by using the epipolar properties. Moreover, in [11] an approximate rectification scheme for the multiple case is also proposed to reduce the back-projection and re-projection calculations.

The color consistency cost function given in 2.13 can not be expressed or approximated analytically in a simple way. Hence, the search space for the depth assignments should be sampled. In [31], the 3D space is sampled as depth planes parallel to the image plane of the reference camera as shown in Figure 2.9 and

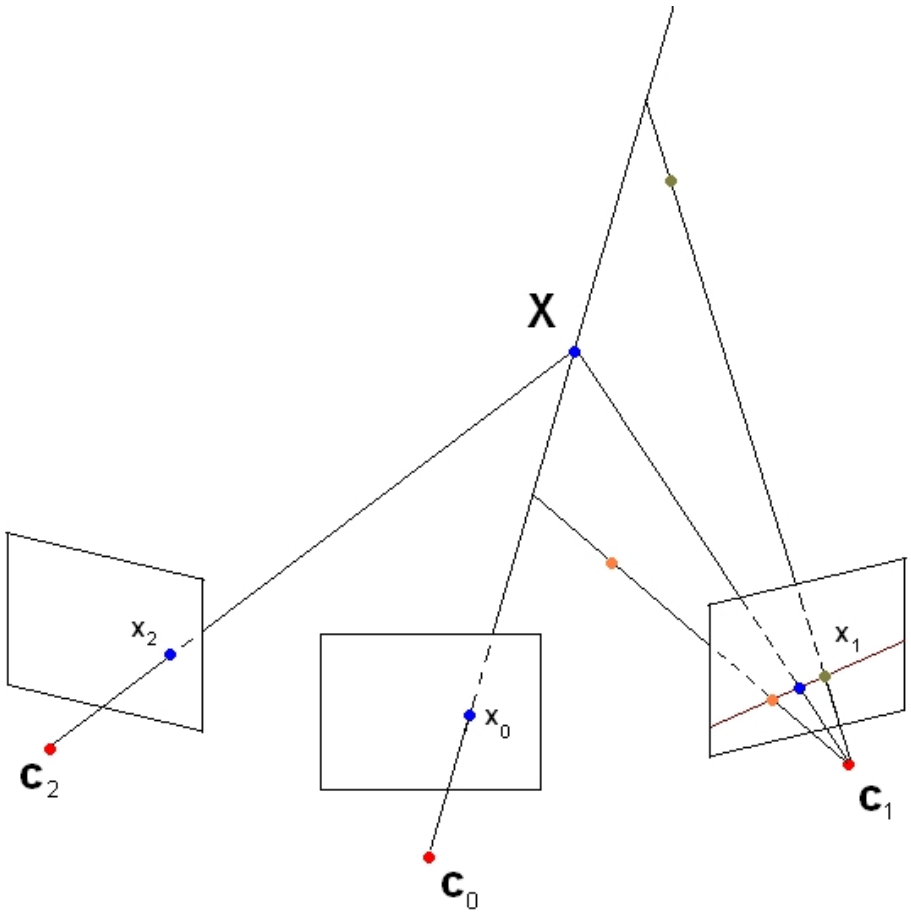

Figure 2.8: Correspondences of a pixel for a depth value

this procedure is usually called as *plane sweeping* [31]. A regular sampling of depth planes may result an irregular sampling on the image planes according to non-linear mapping between the correspondences. In order to reduce the irregular sampling on image planes, the depth planes can be selected to satisfy only one pixel shifts on the epipolar line of another image.

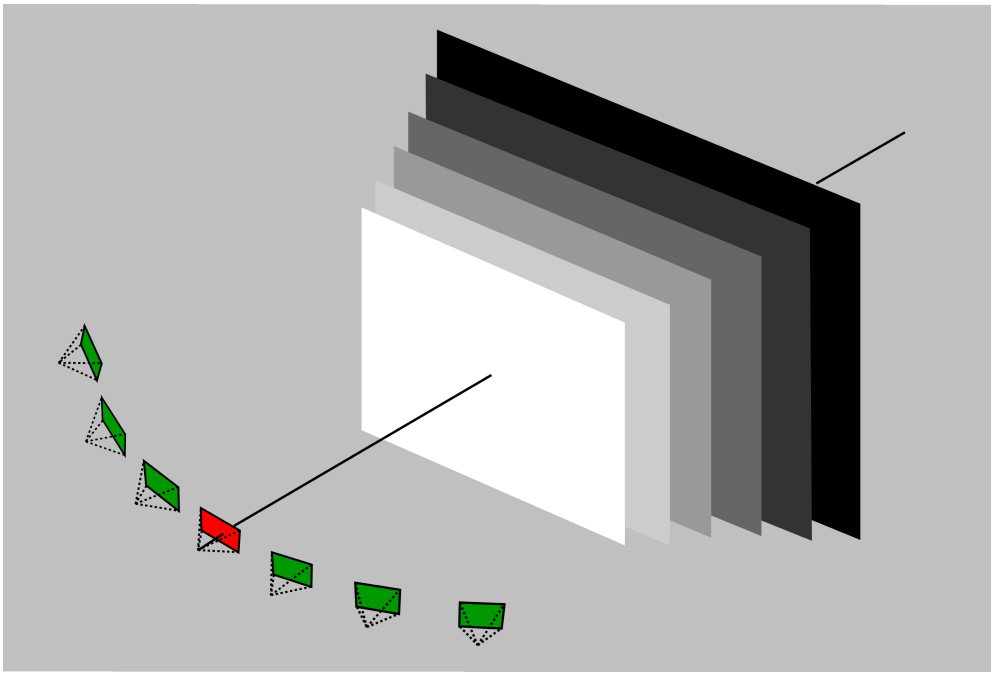

Figure 2.9: The parallel depth planes sampling the 3D space

2.5.2.2 Smoothness Constraint

For imposing the smoothness constraint on the solution, defining a 4- or 8-neighborhood for the MRF model is sufficient. The second order clique energies of the defined neighborhood are set to penalize the depth differences between neighboring pixels.

The undesired effect of smoothing constraint is the loss of sharp depth discontinuities across the object boundaries. For preserving the depth discontinuities of the scene, solving two MRFs jointly for depth and edge maps can be used as in [32], [26]. A more fundamental approach is to saturate the smoothing cost function beyond some defined depth differences. In [26], a robust smoothing cost

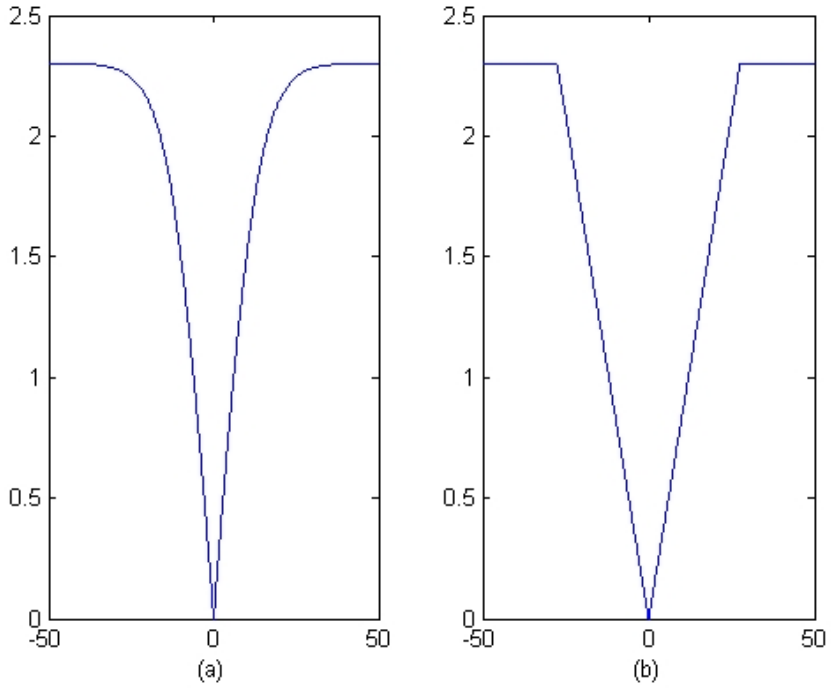

Figure 2.10: (a) Smoothing and (b) linear cost functions for cliques

function given in 2.14, that is derived from a total variance model is used. In [33], a linear cost function with a cut-off is used for practical reasons (see Figure 2.10).

$$V_{C_1}(x_0, x_1) = -\log(\ (1-e) \cdot \exp(\frac{-|d_{x_0} - d_{x_1}|}{\sigma}) + e) \tag{2.14}$$

In general, the object boundaries coincide with the color discontinuities. The smoothing effect across the object boundaries can be suppressed by using the color derivatives of the image. A non-homogenous MRF model, in which the clique energy functions scaled inversely with the color derivative of that neighborhood, favors the smoothness only on similar colored regions.

2.5.3 The Solution of the MRF Model

The Gibbs energy term, stated in 2.12, should be minimized by the dense depth field configuration to be estimated. However, the optimization problem can not be solved analytically or by classical approaches, as it is an *NP-hard* problem [34]. For the solution of the MRF problems, there are many proposed methods in the literature. These methods can be grouped into two main set as stochastic and deterministic [18].

The stochastic approaches are mainly based on a decision depending on a random process result to update the configuration of the field. The fundamental stochastic approaches are Simulated Annealing (SA), Gibbs sampler, and the Metropolis algorithm [18]. The randomness of the update scheme prevents to stick into local minima. In fact, Geman and Geman proved the convergence of the simulated annealing to the exact solution in infinite iterations [23]. However, for the problems like dense depth estimation, the number of depth labels is large and the number of possible configurations increases in an exponential order. Hence, the stochastic approaches are impractical for problems having a large label set.

For the deterministic methods, the iterated conditional modes (ICM) [18], [17], and new approaches developed for inference problems should be mentioned. The ICM method is similar to simulated annealing, except that ICM does not contain any randomness. A much more sophisticated configuration update method is proposed by using graph modeling for the MRF in *graph cut method* (GC) [19].

Another well-developed approach is *belief propagation* (BP) [20], which handles MRF as an inference problem. ICM and BP methods for the dense depth estimation are introduced in the next sections.

2.5.3.1 Iterated Conditional Modes (ICM)

ICM is one of the most widely used approaches for the MRF solution as it is practical to implement and gives satisfactory results for many applications. The main idea of this method is to update the elements of the random field one by one with the minimum cost contributing configuration. The contribution of a field element is calculated by the sum of the clique functions which contain that particular field element as a variable.

The pseudo code of the ICM algorithm for the dense depth estimation problem is given below:

1. `If an initial estimate is available`
    - `set the depth field X with the initial estimate`

      `else`
    - `set X with a random configuration.`
2. `Set the FieldChangeFlag to 1.`
3. `While FieldChangeFlag is equal to 1`
    - `set FieldChangeFlag to 0.`

- `Visit all the field elements in a desired order and do the steps from (a) to (c) for every element.`
  a. `Calculate the cost contribution of the field element` x, $u(x) = V_{C_0}(x) + \lambda \cdot \sum_{x_1 \in C_1}(x, x_1)$ `, for every depth label.`
  b. `Set the` $X_{update}(x)$ `to the depth label which contributes the minimum cost.`
  c. If $X_{update}(x)$ is different from X(x), set FieldChangeFlag to 1.
- `Set` X `to` X_{update}.

Similar to gradient-descent methods, the solution is updated with a configuration, having less Gibbs energy in ICM approach. Since the Gibbs energy function, which is formulated for the dense depth estimation problem, is a non-convex one, the solution of the ICM method may stuck in a local minimum according to the initialization of the depth field. In order to tackle the initialization problem a coarse-to-fine scheme may be used [35], [33]. Defining the elements of the coarser depth fields as image blocks exploits the advantages of block matching algorithms in the color consistency cost calculation.

The main difference of the stochastic approaches, such as SA and GS from ICM, is that they might update the solution with a configuration having a higher Gibbs energy. This property avoids to get stuck in local minima at stake of increasing the total iteration number necessary for the convergence. In GC method,

similar to ICM, the field is updated with a less costly configuration, while the update of the field is not achieved one by one, but by using a graphical model whose multiple entries are changed at once. The details of the GC method can be found in [19].

2.5.3.2 Belief Propagation (BP)

The belief propagation method is widely used to solve the inference problems arising in statistical physics, computer vision, error-correcting coding theory and artificial intelligence. The inference problems are generally represented by graphical models, such as *Bayesian networks* [36], MRFs, or a *factor graphs* [37]. In fact, all these graphical representations can be converted to each other quite easily [20], since they are equivalent.

BP algorithm is a tool for estimating the marginal probabilities of the elements of the MRF. The joint probability density function of the MRF can be written in terms of a factorized form, as given in 2.15 by defining ϕ_i and $\psi_{i,j}$ functions, as the exponentials of the negated first and second order clique functions, respectively, and with a normalization constant Z.

$$p(X) = \frac{1}{Z} \prod_{x_i \in I} \phi_i(x_i) \cdot \prod_{x_j \in \partial x_i} \psi_{i,j}(x_i, x_j) \tag{2.15}$$

A small portion of the factor graph representation of 2.15 is illustrated in Figure 2.11-a. In this figure, the circles denote the MRF elements as variables

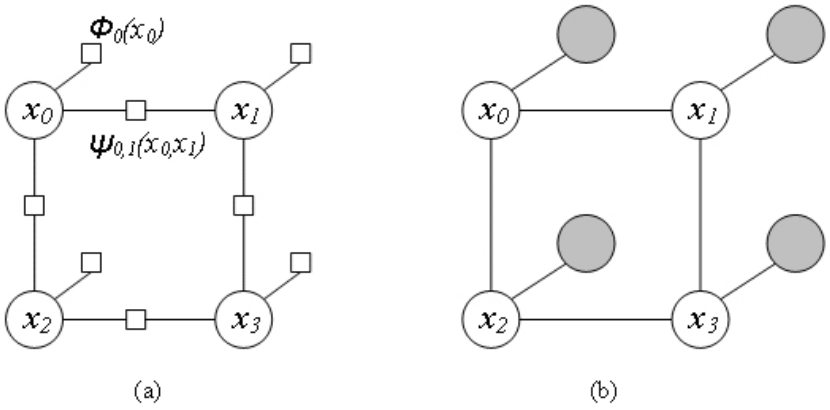

Figure 2.11: (a) Circles denote the MRF elements as variables and squares represent the factoring functions having variables as the MRF elements they are connected to (b) The equivalent graphical representation of the pairwise MRF

and the squares represent the factoring functions having variables as the MRF elements they are connected to.

The equivalent graphical representation of the pairwise MRF is shown in Figure 2.11-b, in which the unfilled nodes represent the field elements, hidden nodes, and the filled ones represent the observed nodes. The observed nodes put condition on the hidden nodes, for which they are connected to, proportional to the observed local evidences, which are ϕ functions in the factor graph representation. For the inference between the hidden nodes a message receiving-sending scheme is introduced in the BP algorithm.

A message from hidden node-i to hidden node-j is a vector, whose dimension is equal to the number of possible states of node-j, and is denoted as $m_{i,j}(x_j)$.

In the BP algorithm, the belief at a hidden node x_i is defined as the marginal probability density of x_i, $p(x_i)$, and it is proportional to the product of the local evidence at node x_i, $\phi_i(x_i)$, and all the messages coming into x_i. The k parameter in 2.16 is a normalization constant for summing up the beliefs to 1.

$$p(x_i) = k \cdot \phi_i(x_i) \prod_{j \in \partial i} m_{j,i}(x_i) \tag{2.16}$$

The messages are calculated recursively with the message update rule of the BP algorithm that is given as [20]

$$m_{j,i} \leftarrow \sum_{x_i} \phi_i(x_i)\psi_{i,j}(x_i, x_j) \prod_{k \in \partial i \setminus j} m_{k,i}(x_i) \tag{2.17}$$

The $\psi_{i,j}$ function is a compatibility function between the neighboring nodes i and j, that is derived from the second order clique functions, as mentioned before. This message update rule is also known as the *sum-product algorithm* [36], since it is the summation of some product terms. The algorithm is initialized with messages having a uniform distribution. It is shown [38] that the beliefs of the nodes of a graph, with no loop, converge to the exact marginal probabilities by the BP algorithm. For the graphs with loops, the BP algorithm does not provide the exact inference, but still a good approximation, which is proved to be a *Bethe approximation* in [20], is achieved. After sufficient iterations or convergence, the belief of each field element is computed and the every element is assigned to most probable label for the solution of the MRF.

The BP algorithm can be briefly summarized as a method that utilizes local computations to understand the inference on a much more complex global models. The messages generated with simple local computations flow on the network in a source to drain scheme. In this manner, the messages from the source are summarized according to the network model and send to the destination drain to make a global inference.

Different from ICM, the BP algorithm handles all the possible configurations in computation so that the BP algorithm does not suffer from the initialization problem, as much as the ICM algorithm. For speeding up the convergence of the BP algorithm, the same coarse-to-fine scheme, mentioned for ICM, is also widely used [33].

The message update rule in 2.17 involves many multiplications, which becomes impractical for the cases having many labels to assign. In order to tackle this issue, the message update rule can be modified as [26]:

$$m_{j,i} \leftarrow \max_{x_j} \phi_i(x_i)\psi_{i,j}(x_i, x_j) \prod_{k \in \partial i \setminus j} m_{k,i}(x_i) \tag{2.18}$$

Moreover in [33], the message update rule is not performed on the probability domain, but on the cost domain, in order to use summations, instead of multiplications and a linear second order clique function, presented in Figure 2.10 is used for calculation efficiencies. In the literature, these modified BP algorithms are called as *max-product* [26] and *min-sum* algorithms [33]. The message update

rule of *min-sum* algorithm is given in 2.19.

$$m_{j,i} \leftarrow \min_{x_j} (-\log \phi_i(x_i))(-\log \psi_{i,j}(x_i, x_j)) \sum_{k \in \partial i \setminus j} m_{k,i}(x_i) \tag{2.19}$$

2.6 Simulations on Dense Depth Estimation

There are three main objectives in the experiments explained in this section. The first one is to compare the performances of two MRF solving approaches, well-known ICM and a contemporary one, BP. The second objective is to compare the dense depth estimates from two view stereo against that of the multi-view stereo. Finally, the last aim is to observe the effects of the smoothing constraint on the dense depth estimates. All these experiments are conducted by using fully calibrated views.

In order to compare the ICM and BP algorithms, a stereo image pair, which is presented in Figures 2.12 a and b, is used. The special characteristic of this particular stereo image pair is that it has many feature points which makes correspondence finding much easier. For the fair comparison of the ICM and BP methods, the same number of depth labels are used, exactly the same clique functions are defined, no initial estimate is utilized and the same coarse-to-fine scheme is implemented. Based on the simulations, given in Figures 2.12 c and d, the depth estimate results of the BP algorithm gives much more realistic results. It should also be noted that, in [39], the benchmark results are given for various stereo disparity estimation algorithms, and BP algorithm is reported as one of

the best solutions.

For the comparison of depth estimation from two view and multi-view sources, *Microsoft Breakdancer* sequence is used. The *Breakdancer* data consists of 8 videos captured in a synchronized fashion by fully calibrated cameras. All 8-views for an arbitrary time instant are given in Figure 2.13.a. The depth estimates of the mid camera from two views and 8 views are acquired by the min-sum BP algorithm and presented in Figures 2.13 b and c, respectively. In the min-sum version, the linear second order clique energy, which is shown in Figure 2.10.b, is used for computational efficiencies as mentioned in [33]. In Figures 2.13 d and e, the dense depth estimates of the standard BP algorithm and the ICM are compared. The results show that using multiple images makes the color consistency constraint much more robust. Another observation is that the min-sum BP results are preferable not only for computational efficiency but also the quality of the estimates, as well.

The smoothness of the dense depth map solutions can be controlled by varying λ parameter of 2.12, that is the weight of the smoothness constraint in the MRF model. The dense depth estimates for the increasing λ values are presented in Figures 2.14 a to d. The smoothness of the estimates is increasing as expected.

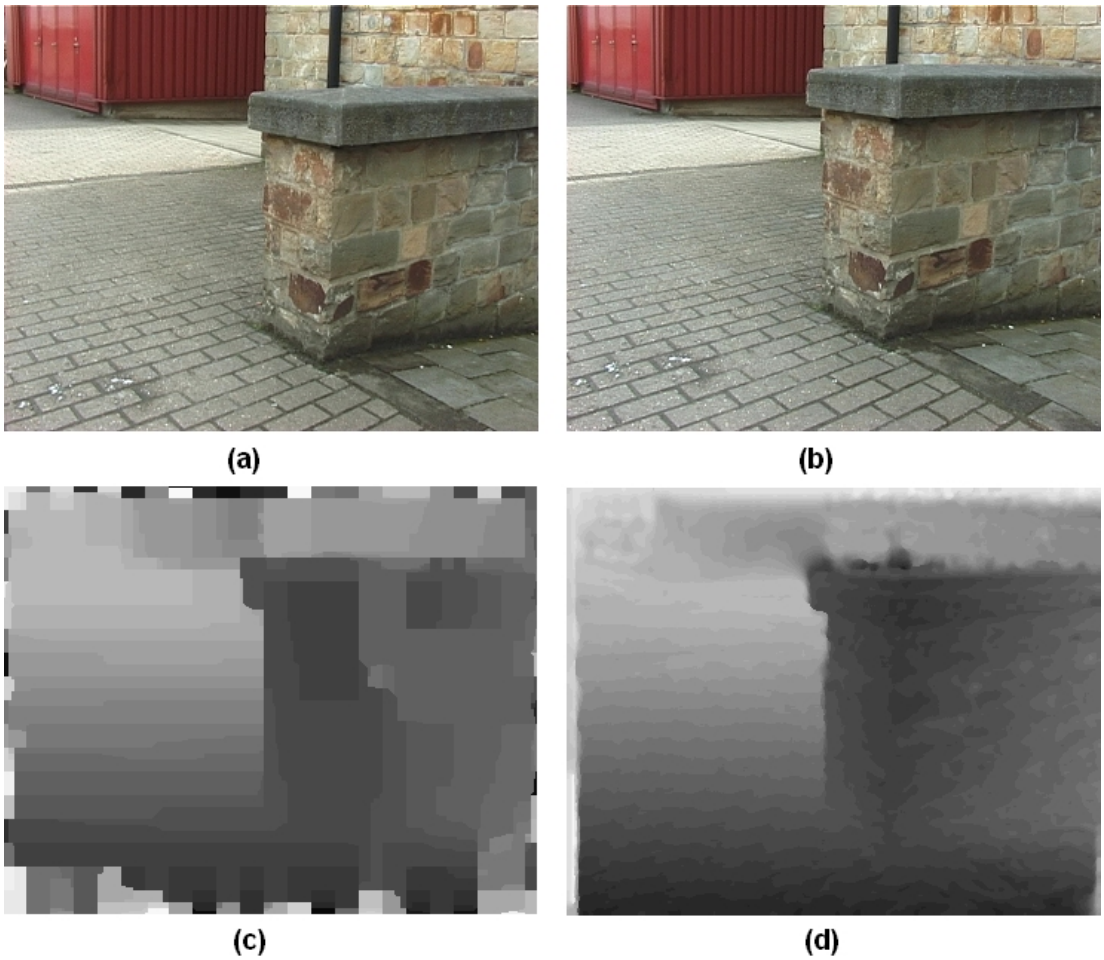

Figure 2.12: ICM and BP comparison for stereo case. Left and right image pairs are shown in (a) and (b). The dense depth estimates of ICM and BP are given in (c) and (d) respectively. The darker parts of the dense depth estimates represent the nearer parts of scene.

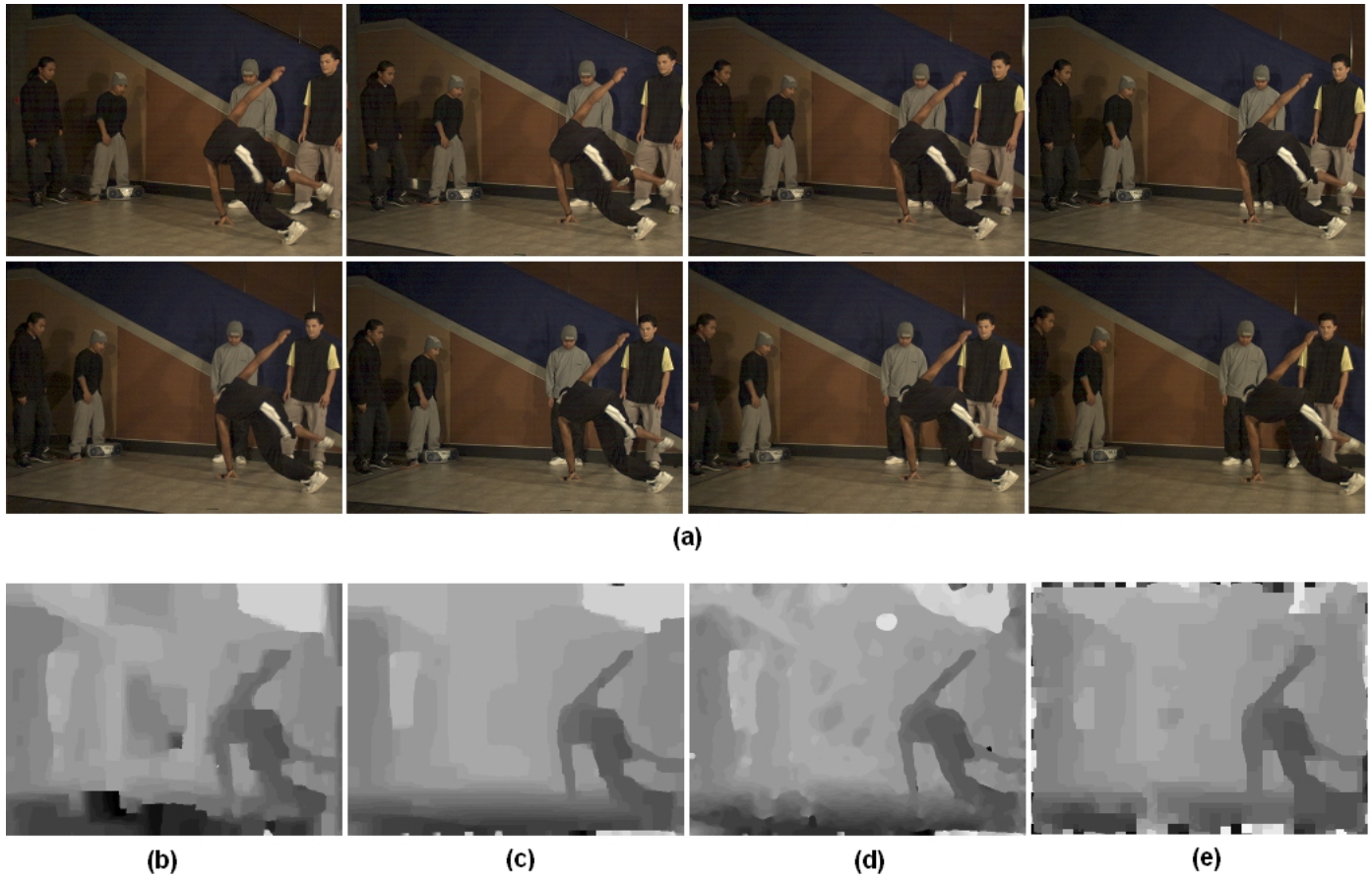

Figure 2.13: Multi-view dense depth estimation simulations. The 8 views of the *Breakdancer* (a). Dense depth estimation from two views (b) and 8 views (c) using min-sum BP method. The dense depth estimates of sum-product BP (d) and ICM (e) for *Breakdancers*.

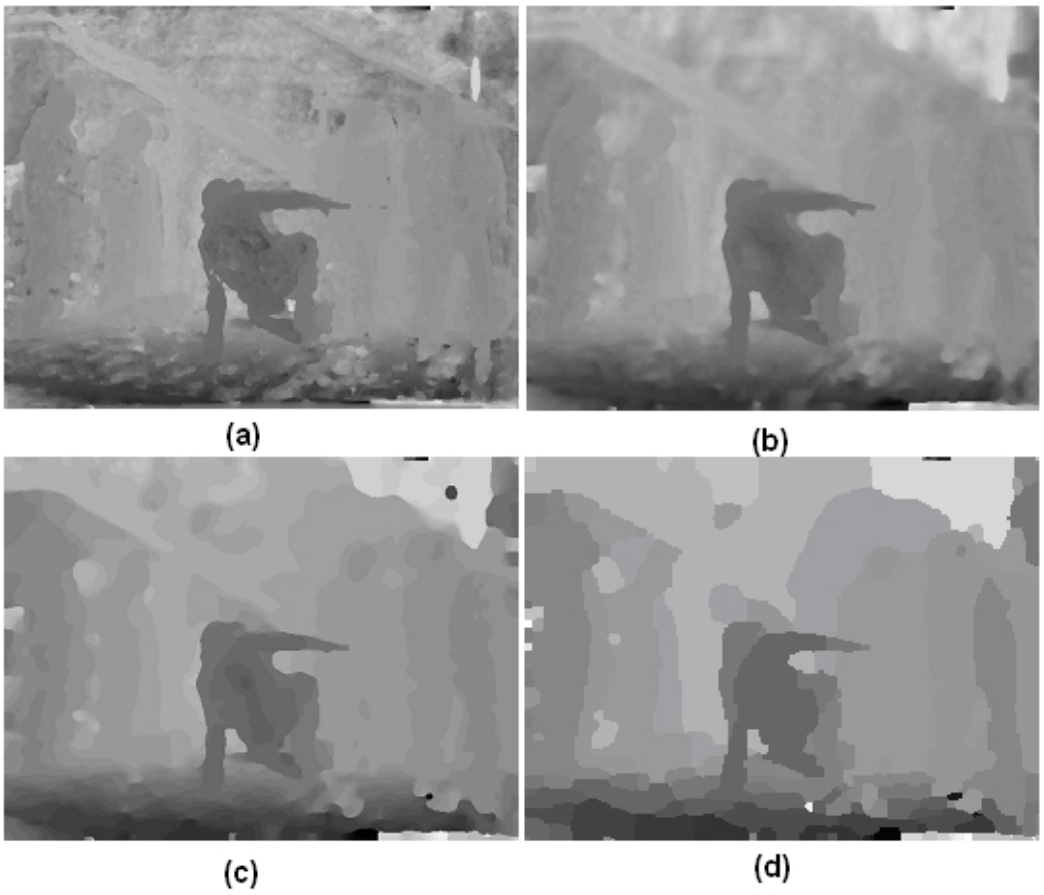

Figure 2.14: Evolution of the dense depth estimates while the smoothness constraint gets dominant in the MRF model from (a) to (d).

CHAPTER 3

NOVEL VIEW GENERATION

In the literature, estimating an arbitrary view of a virtual camera by using the given image sources and any other available related information is denoted as *novel view generation*. In this chapter, a 3D warping algorithm will be explained for the solution of the novel view generation problem. Firstly, the 3D scene representation part of the problem will be mentioned and the main approaches in the literature will be briefly explained. In the 3D warping algorithm, a hole-filling method will also be introduced to handle the artifacts caused by the occluded regions. Lastly, the fusion of two novel views will be studied to increase the visual quality of the novel views.

3.1 3D Scene Representations

The increasing interest in immersive media and projects, such as 3DTV broadcast, fuels the search of standardization in the representation of the 3D data. The

main issue in the representation problem is to find a description favoring high quality rendering and efficient compression for coding. Hence, the representation problem is mainly in the domain of research activities of computer graphics and data coding disciplines. At present, the main 3D scene representation methods can be classified as point-based, surface-based, volume-based and image-based methods.

In surface-based representation, the bounding surfaces of the objects in the scene are mainly considered. The surfaces might have any arbitrary shape and they generally do not have an analytical form. The polygonal meshes and NURBs [40] are the most widely used elements for such surface representations. The major advantage of a surface-based representation is the fact that the present rendering infrastructure mainly supports the polygonal meshes. However, the mesh structure has serious shortcomings in terms of coding perspective.

In recent years, as an alternative to mesh based rendering pipeline, point primitives have received growing attention in computer graphics. The 3D scene is represented by a point cloud, which is generally sampled non-uniformly and each point has some attributes, such as color, surface normal and splat size. By using these attributes and connectivity of the point cloud, the scene is rendered. In [41], a fully complete 3D video system, from capturing to coding, is proposed based via point-based representation.

For the visualization of the 3D data encountered in scientific and medical

areas, the volumetric representation has been used for many years. In order to extract and represent the 3D scene structure, the volumetric approaches, like *voxel coloring* [42] and *space carving* [43], are also utilized in computer vision solutions.

The last approach to be mentioned is the image-based representations. A collection of conventional 2D images and a model for the scene geometry is used to represent the scene. The major advantage of the image-based representation is the independency of the scene rendering problem from the complexity of the scene geometry, that is a common problem for the representations based on some geometric primitives. The rendering issues of the image based representation will be mentioned in the next section.

At present, the solutions to the representation problem are quite application dependent. However, for the immersive media broadcast, the image-based representation seems to be more compatible with the current infrastructure. Moreover, the rendering quality of the image-based representations is very promising.

3.2 Image-Based Rendering (IBR)

The image-based rendering (IBR) term is used for various approaches whose common identity is to exploit a collection of images. The utilization of the source images in novel view generation may vary abruptly from geometric models to statistical models. However, the IBR methods can be generalized in terms of

utilization of interpolation and pixel reprojection during novel view rendering which is completely different from conventional rendering pipelines. The promising features of the IBR are natural looking novel views by using real pictures and capable of handling difficult geometric structures, such as hair, fur, leaves, in a fixed computational complexity.

Two fundamental and breakthrough methods in IBR are the *light field rendering* [44] and *unstructured lumigraph* [45]. The main idea of the unstructured lumigraph is finding possible sources among the collection of images for each pixel of the desired novel view and weighting the sources with respect to the available constraints, like orientations of the source cameras and information on the scene geometry. The light field rendering is a special case of unstructured lumigraph, in which the source cameras are oriented as a matrix on a plane and the scene is assumed to have a planar geometry.

Another IBR method, which is strongly based on the scene geometry, is the *layered depth images* (LDI) [46]. LDI method should also be considered as a novel 3D scene representation method. By using the dense depth map of each image in the collection, the color and depth information is gathered into layered image plane of a virtual camera whose field of view encompasses the whole scene. A pixel of a LDI holds the color and depth information and these pixels are ordered with the intersection order of the projection rays with the scene. The desired novel views are rendered by using the visibility information encoded in

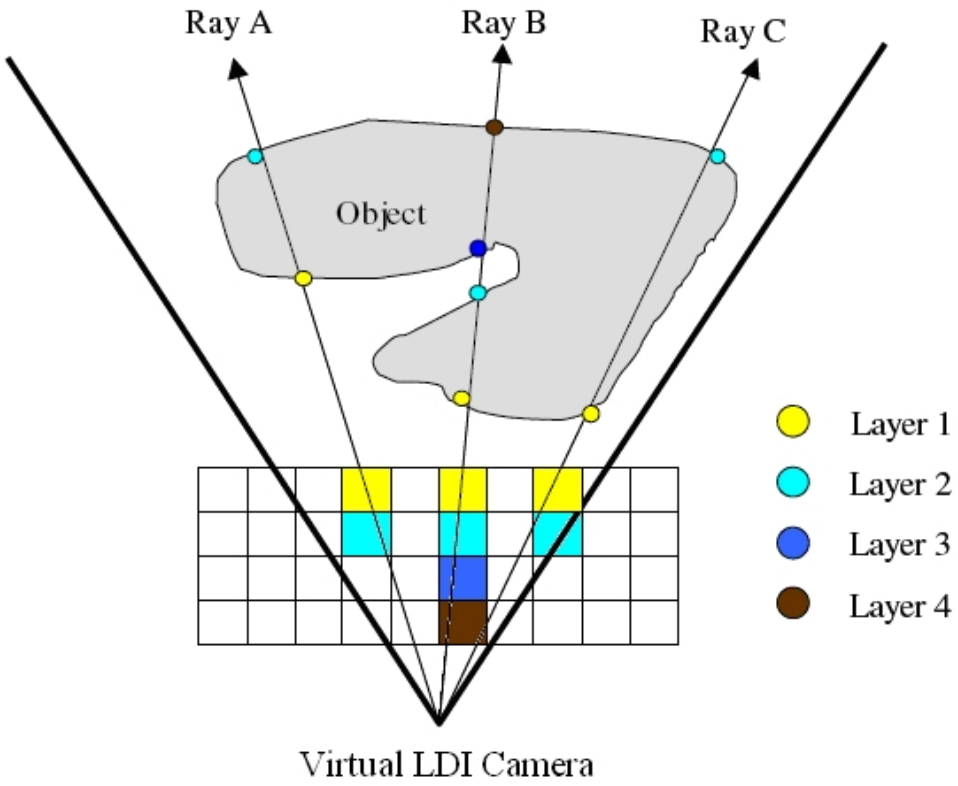

Figure 3.1: Generation of a layered depth image

the layers. The generation of a LDI is illustrated in Figure 3.1

There are more weakly geometry-based IBR methods which are also proposed in the literature. In [47], the ordering constraint on the epipolar lines are used to generate intermediate novel views from two and three images. A different IBR approach is proposed in [48] by the motivation that a novel view should have similar texture statistics with the images in the collection. A novel view is generated by using the textures patches in the source images considering the epipolar constraints and texture statistics.

In the next section, an IBR method, which is called 3D warping, will be explained in detail. The 3D warping method can be evaluated as a variant of unstructured lumigraph that uses the depth information of the source images. The distinguishing property of the 3D warping method is that it is developed for just one image available cases.

3.2.1 3D Warping Algorithm

3D warping can be briefly explained as a mapping from the image plane of the source camera to image plane of the virtual camera. The 3D phrase is emphasized, since every pixel on the source image follows the path from source image plane to 3D world, and then to novel image plane. As long as the dense depth map of the source image and the calibration information of the cameras are known, the mapping can implemented for a point, as back-projection, rotation, translation and re-projection in this order.

Naive 3D warping result for a frame in the *Breakdancer* sequence is shown in Figure 3.2 According to the order of the pixel mapping, the regions, which should not be observed from the viewing direction of the novel camera, are visible in this novel view. For example, the leg of the man, that is indicated with the red circle in the same figure, should be occluded by the arm of the dancer in front. The green lines are other artifacts due to pixel-by-pixel mapping. And lastly, a forward mapping, such as source to destination, needs rounding of the exact

Figure 3.2: A naive 3D warping result with the resulting artifacts, denoted by green regions.

positions of the mapped points to some discrete pixel locations.

In order to remove the resulting artifacts of such a naive approach, reconstruction of 2D manifolds (surfaces) in 3D world and projection of these manifolds into desired image plane by using a depth buffer, is proposed in [49]. The forward mapping artifacts might also be removed by using an interpolation on the topology of the manifold.

The reconstruction of the 2D manifolds in 3D space can be easily obtained by fitting a mesh on the reference image. The regular grid of the pixels provides

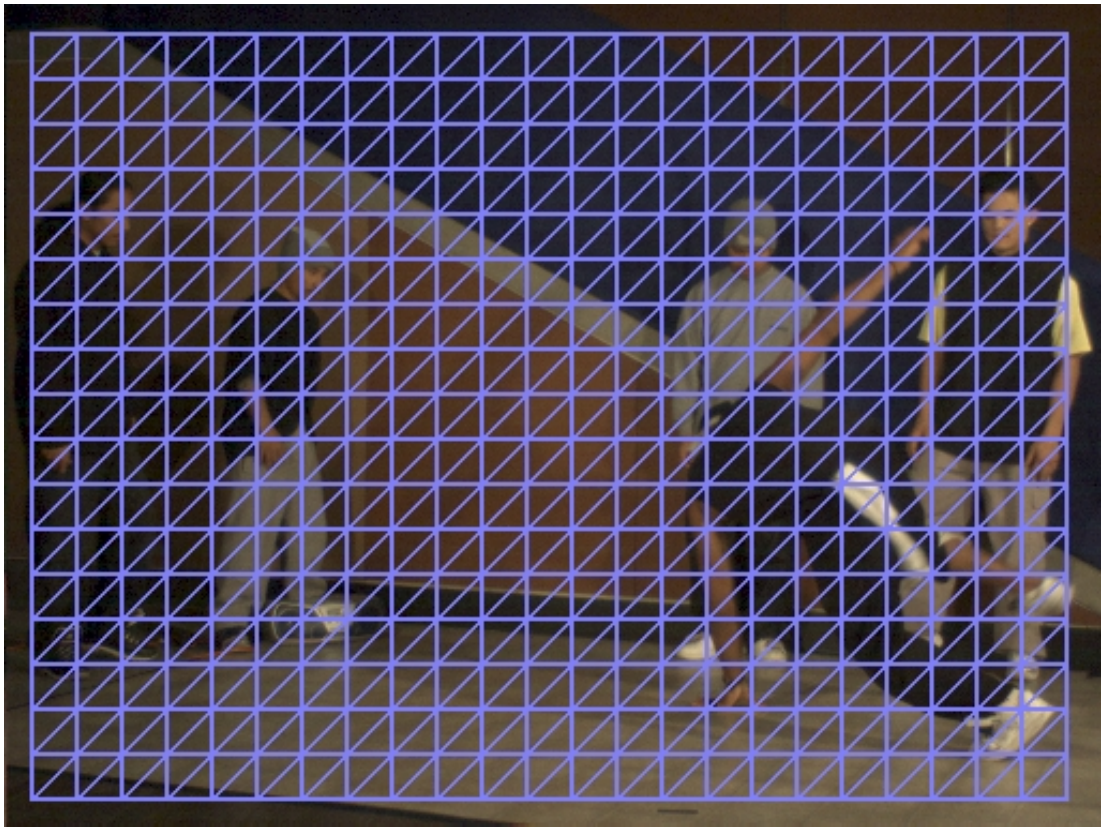

Figure 3.3: Illustration of the regular mesh on the image plane

a natural mesh structure. Back-projection of the pixels to the 3D space, as the vertices of the mesh, whose link list is derived from the regular image grid defines the desired 2D manifold. An illustration of the regular mesh on the source image is shown in Figure 3.3.

A serious drawback of using a regular mesh is the construction of the whole scene as a single connected surface. In fact, the real 3D scenes are generally composed of various disconnected surfaces. Imposing connectedness might result with disturbing interpolation artifacts at the regions, which are occluded in the

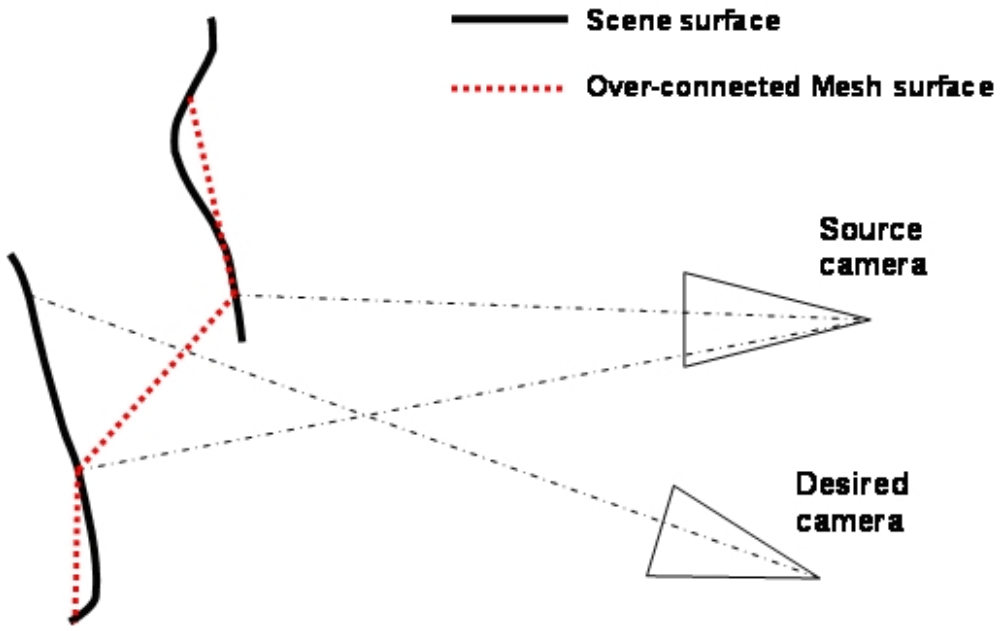

Figure 3.4: Visibility artifacts of an over-connected surface

source image, whereas visible in the novel view. By using a proper distance threshold, the connectedness of the vertices of the 3D mesh should be tested to avoid the over-connectedness of the scene. In the generated novel view, these discontinuities appear, as holes due to occlusions and handling of these occluded regions will be the focus of the next section.

The rendering part of the novel view generation becomes sampling of the 2D manifolds in the 3D space at the crossings of the projection rays of the pixels of the desired image plane. The intensity (color) information of the sample is interpolated from the vertices of the triangle on which it is located. The architectures of the present graphic hardware are designed and optimized for mesh rendering.

Figure 3.5: A novel view generated by 3D warping algorithm

By using a graphics library, such as openGL [50], any novel view can be easily rendered while taking the visibility issues into account by using a depth buffer. In Figure 3.5, the same view, which is generated with a naive approach in Figure 3.2, is obtained by the 3D warping algorithm.

3.3 Hole Filling

According to the estimated, acquired or modeled scene geometry, the generated novel views contain non-rendered regions, which are denoted as *holes*. Such holes

are not visible in the source image due to occlusions. In fact, the corresponding visual information at these hole locations are totally unknown, if one only considers the visual data at the reference view. In order to fill out these holes, some interpolation/extrapolation or patching methods are usually utilized in the literature. Interpolation/extrapolation-based methods fill such holes by using the intensity information of the nearest rendered region to avoid disturbing effects [51], [52]. In patching-based approaches, the texture of the bounding region is pasted onto the holes [53]. In a recent study [54], an elegant PDE-based novel view generation method is also proposed which handles the hole filling problem during the sampling of the scene.

The main strategy of the hole filling algorithms is distinguishing the background section of the boundary for the hole. Since the occluded regions must lay behind some other parts of the scene, it is reasonable to utilize the color or texture information of the background during fill operation. In order to check the background and foreground parts of a hole, the epipolar lines should provide a perfect orientation. An epipolar line, crossing a hole, passes through two neighboring points in the source image. An illustration of this fact is shown in Figure 3.6. Hence, a basic hole filling algorithm can be derived as the extrapolation of the background boundary colors along the epipolar lines on the hole [51].

In [55], it has been proven that knowing only the sign of the epipole on the source image defines an occlusion-compatible mapping order to warp the source

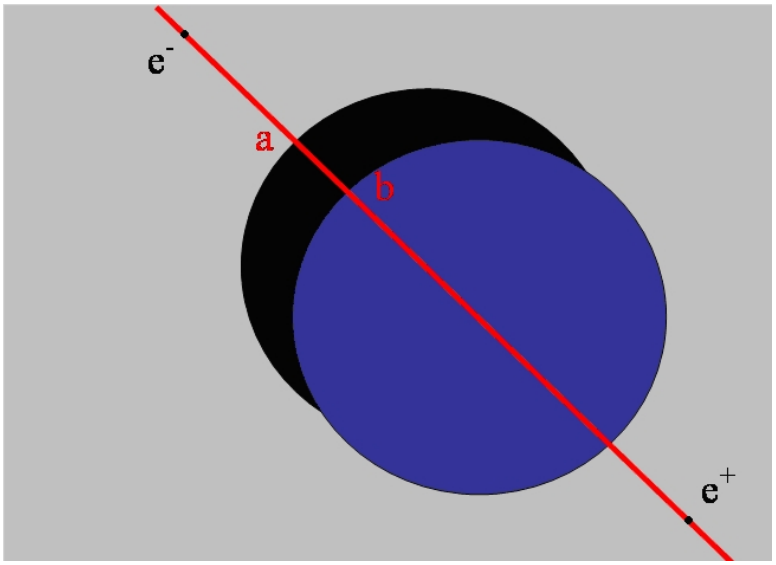

Figure 3.6: The 3D warped view of a blue disc, standing at the front of a gray background. The black region denotes the hole and the red line is an epipolar line crossing the hole. Points a and b are two neighboring points in the source image. Possible positive and negative epipoles are also shown.

image. A sketch of the proof is given in Appendix A. For the positive epipole case, the source image should be mapped towards the epipole and away from the epipole for the negative epipole case. In [49], the occlusion compatible mapping order is used as the hole filling order. The sign of the epipole of the novel view guarantees to fill the holes from background to foreground.

According to the position of the epipole with respect to the visible part of the image plane, there are 9 possible configurations. Three distinct cases are illustrated in Figure 3.8. The remaining configurations are identical to cases in Figure 3.8 b and c. The position of the epipole divides the image into 4, 2 or just 1 parts, which are denoted as *sheets* [55]. In order to be implement in practice,

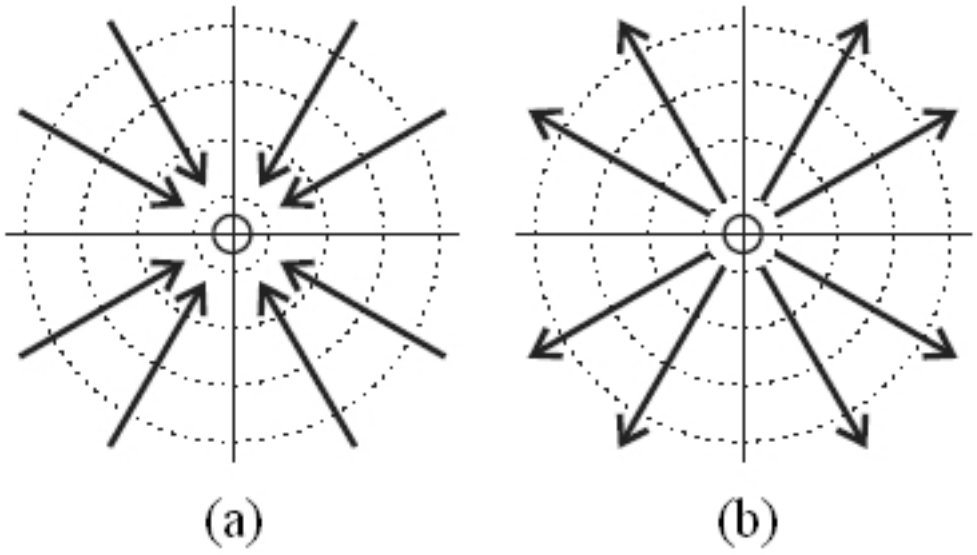

Figure 3.7: The occlusion compatible directions for the positive and negative epipoles are shown in (a) and (b), respectively.

the scanning directions, which should be perpendicular to the epipolar lines, are approximated with three main directions; horizontal, vertical and diagonal. All three cases, illustrated in Figure 3.8 are given for the positive epipoles, that means scanning towards to epipole.

Using this scanning framework proposed in [49], the pixels belonging to the holes are detected and the color of the background pixel in the scanning direction, which are shown in Figure 3.8 as bold arrows, is extrapolated (i.e. copied). The scanning order guarantees that the source pixel for the extrapolation is not an unfilled pixel. Hence the holes of the novel image are filled in just one scan considering the background-foreground distinction.

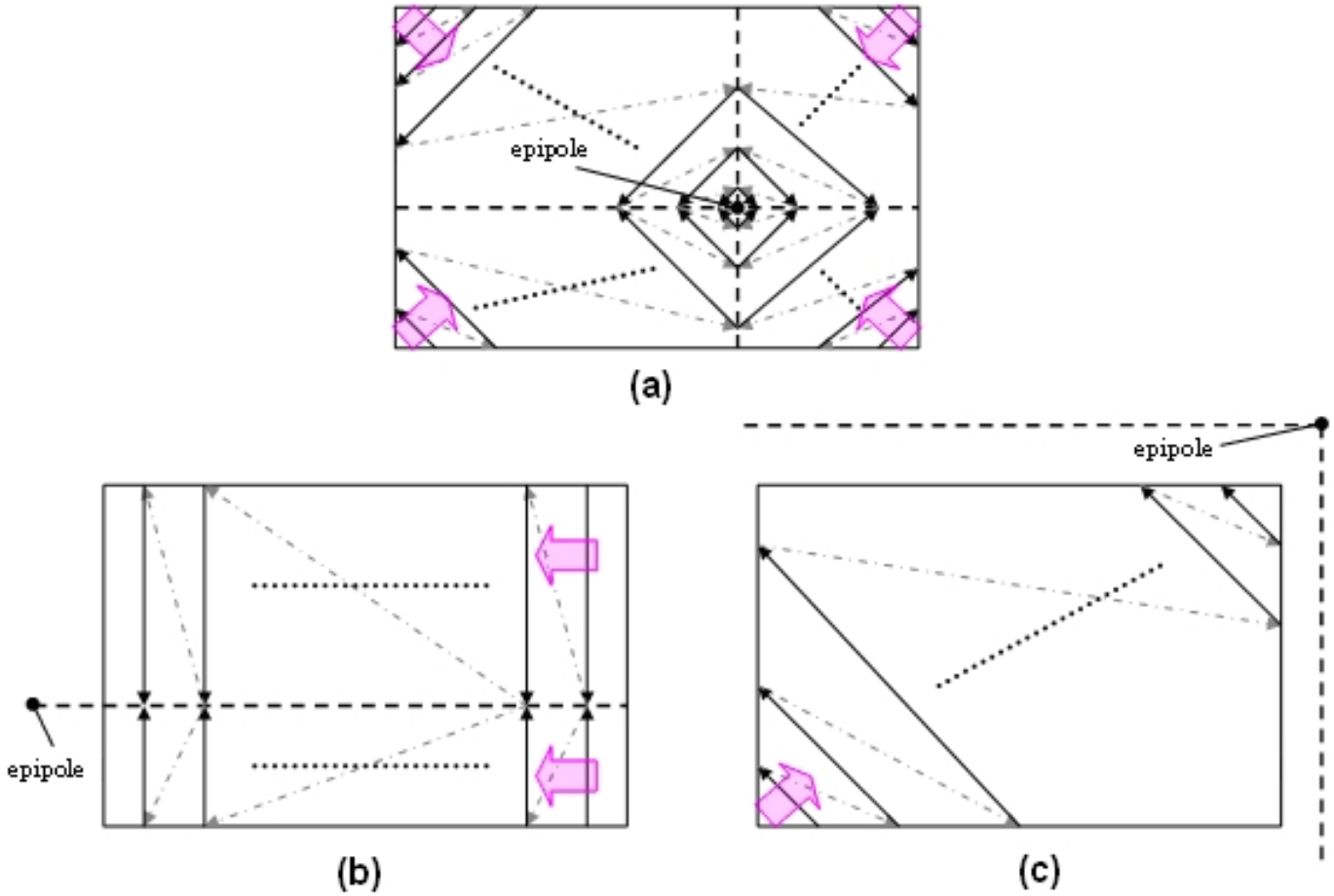

Figure 3.8: The three possible sheet arrangements according to epipole positions with respect to the visible part of the image plane and their resulting occlusion-compatible scanning order for hole filling.

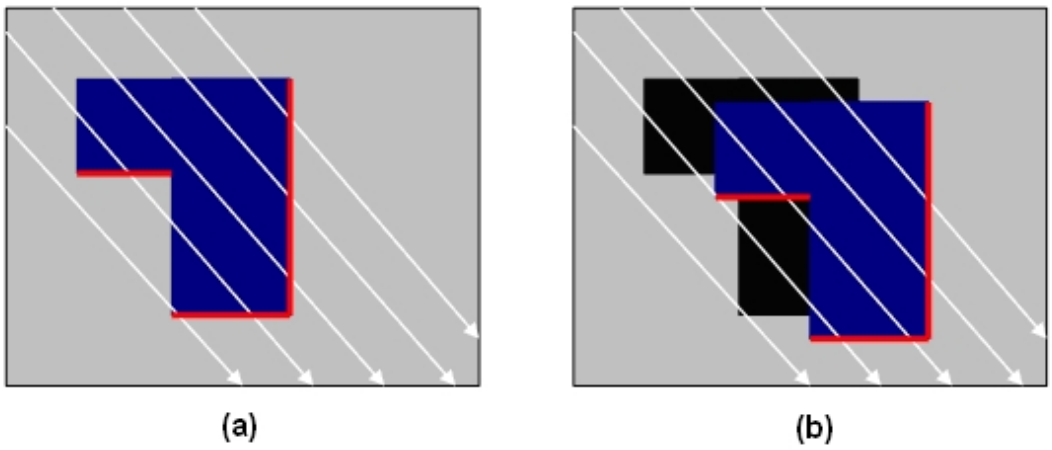

Figure 3.9: In (a) a concave blue object standing in front of grey background and its 3D warp is shown in (b). The red edges are the forward edges with respect to the epipolar direction.

The hole filling method explained above might fail in case of a hole, which is created by a concave foreground object. Figure 3.9.b illustrates an example for this case. In [49], for tackling this problem, *forward edge* of an object is defined on the source image. The object edges are the natural boundaries of the unconnected surfaces of the 3D mesh which is used in 3D warping. The object edges, which pass from foreground to background in the direction of the intensity extrapolation, utilized during hole filling, are defined as *forward edges* [49]. The hole filling algorithm can be modified easily by checking the source of the extrapolation, whether it is from a forward edge or not, to avoid foreground extrapolation.

A novel view before and after the application of hole filling is presented in Figure 3.10. The experimental result concludes that the hole filling algorithm

Figure 3.10: (a) 3D warped novel view and (b) result of hole-filling

might handle the occlusions in general, especially on the stationary backgrounds, such as the floor or the wall in the given view. However, the occluded regions of more complex backgrounds, like the man in the view, can not be handled acceptable in terms of the visual quality. Moreover, the extrapolation of the background might also become disturbing in large occluded regions, which are encountered frequently in wide baseline cases.

3.4 Intermediate Novel View Generation

As long as the source image does not contain any information about the occluded regions on the novel view, the hole filling algorithms are limited to generate some predictions. However, if it is possible to increase the number of the view sources, those occluded regions in one of the source might be compensated from another source. An arrangement of two source views, that leaves the novel view as an

intermediate, handles most of the occlusion problems.

The intermediate novel view generation algorithm can be interpreted as a fusion of two hole-filled novel views, considering the occlusions. The algorithm can be summarized as follows.

1. 3D warp the source images to novel view separately.

2. Apply hole filling to each novel views acquired and keep the masks of the occluded regions of each novel views.

3. Repeat the following steps for each pixel on the novel view:

    - If that particular pixel is visible in just one of the novel views copy the color of the visible pixel.
    - Otherwise fuse the color of two pixels by weighting them in reverse proportional to the distances between the camera centers of the sources and the novel view.

The simulation results of the intermediate novel view generation method are given in Figures 3.11 and 3.12 for the *Breakdancer* and *Ballet* sequences, respectively. In order to compare the objective quality of the results, an available image in the multi-view set is selected to be estimated, as the novel view. Two source views and their dense depth estimates acquired by the BP method are used in the intermediate novel view generation. In Figures 3.11 and 3.12, the upper row

consists of left source view, original novel view and right source view in left-to-right order, whereas the bottom row consists of 3D warp of left source view, fused intermediate novel view and right source view in the same order. PSNR values of the intermediate novel views with respect to originals are obtained as 32.44 dB and 28.88 dB for typical frames from *Breakdancer* and *Ballet* sequences, respectively. The reason of the drop in the PSNR value of the Ballet sequence is the inferiority of the dense depth estimate quality, due to severe occlusions in the scene geometry. However, it should be mentioned that the results are quite promising in terms of subjective quality.

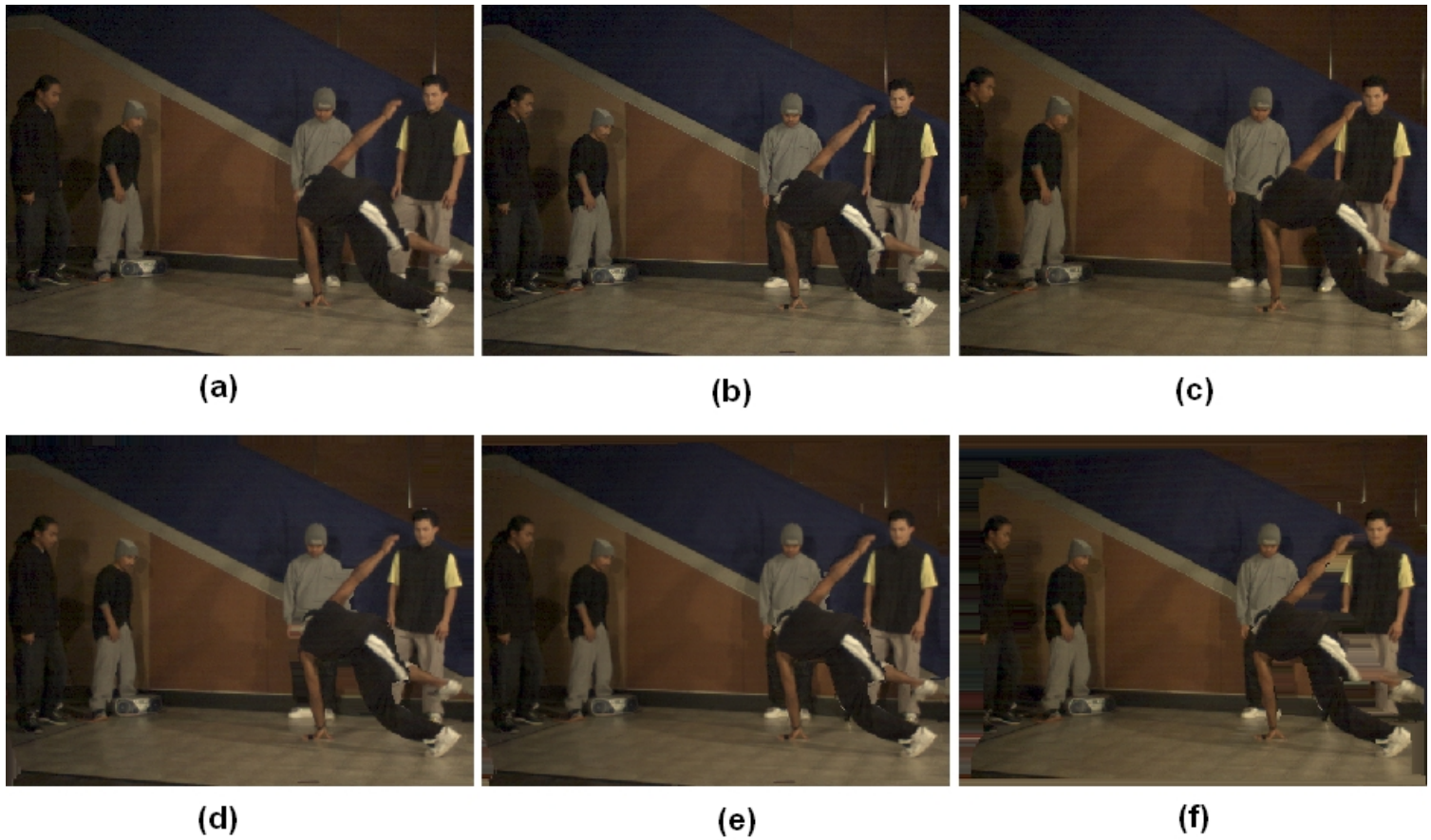

Figure 3.11: Novel view generation result for *Breakdancer*. The left and right source images are shown in (a) and (c). The original of the generated novel view is given in (b). The 3D warped images of the left and right sources are shown in (d) and (f) respectively. The final fused intermediate novel view is given in (e).

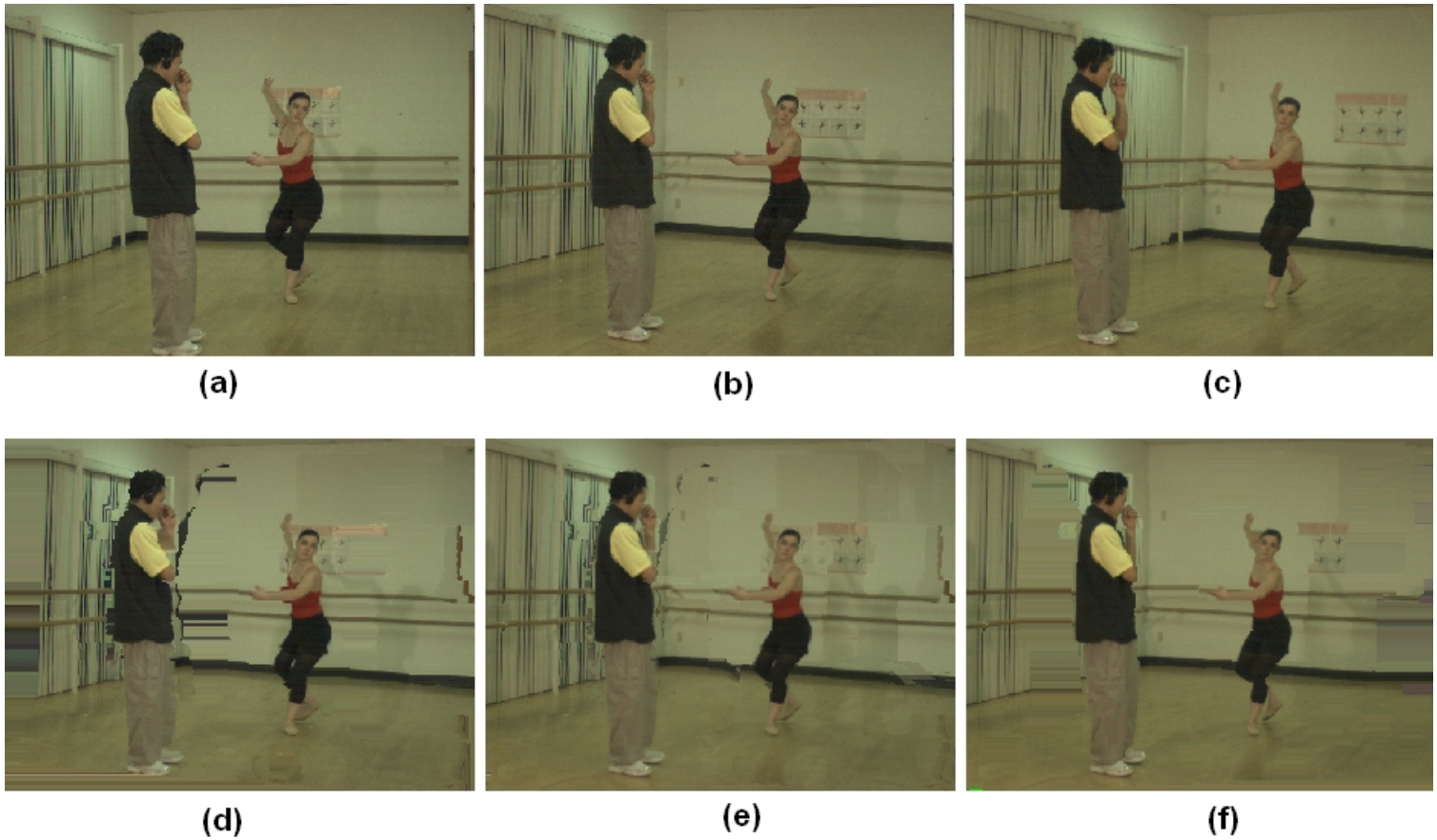

Figure 3.12: Novel view generation result for *Ballet*. The left and right source images are shown in (a) and (c). The original of the generated novel view is given in (b). The 3D warped images of the left and right sources are shown in (d) and (f) respectively. The final fused intermediate novel view is given in (e).

CHAPTER 4

MULTI-VIEW VIDEO CODING

In this chapter, the recent research on compression of multiple video sequences from cameras (more than two) viewing the same scene, that is known as *Multi-view Video Coding* (MVC), is explained and a novel dense depth-assisted MVC approach will be proposed. The performance of the proposed MVC method is analyzed in a special test bed, which will be explained in detail. The chapter starts with a summary of the video coding fundamentals with emphasis on the recent ITU H.264 (ISO MPEG-4 version 10) video coding standard. Next, the MVC problem is defined and the ongoing standardization activities are summarized, as a research survey for MVC. And lastly, a novel approach, which utilizes the tools introduced in the preceding chapters, will be proposed with some simulation results.

4.1 Video Coding Fundamentals and H.264 Standard

Current communication channels necessitate compression of a typical video data for its transmission. The video data can be interpreted as a 3-dimensional data, which consist of 2-dimensional spatial component and a single dimensional temporal component. Unless the video data is a random noise, there exist many spatial and temporal redundancies which play a vital role in its compression. The spatial redundancies arise from the visual coherency of the objects in the scene, whereas the temporal redundancies arise from the smooth continuity of the object and camera motions in time. Another source of redundancy is the statistical one, which arises from the representation of the information before compression.

There are two main organizations developing video coding standards, that are ITU-T Video Coding Experts Group and ISO/IEC Motion Picture Experts Group (MPEG). In 1991, the first video coding standard, H.261 [56], is introduced by ITU-T for videoconferencing and it is followed by H.262 [57] and H.263 [58] standards. MPEG-1 [59] and MPEG-2 [60] are the two standards introduced by MPEG group and they are widely used in the consumer products, VCD and DVD, respectively. Other widespread applications of MPEG-2 can be listed as cable TV and High Definition TV (HDTV). Afterwards, the joint work of these two organizations, called as *Joint Video Team* (JVT), developed the H.264 standard.

All contemporary video coding standards exploit the spatial, temporal and

statistical redundancies in the video data. The architectures of the aforementioned coding standards have a common structure, since they are built upon the preceding standard. Most of these standards follow the industrial requirement for being backward compatible to their parent standard. In order to explain the methodologies for exploiting the redundancies existing in the video data, H.264 standard, which can be assumed as the-state-of-the-art in video coding, is preferred [61], [62], [63]. It is noteworthy that the improved features of H.264 have almost doubled the compression performance against MPEG-2 standard.

4.1.1 Major Features of H.264

The encoder block diagram of the H.264 standard is given in Figure 4.1. The frames of video are split into 16x16 pixel blocks which are called as *macroblocks*. The macroblocks encoded in a raster scan order from top left to bottom right of the frame. Each macroblock consists of three components which are luminance, Y, and chrominance channels, Cr and Cb. As human vision system is less sensitive to chrominance channels, they are usually subsampled by a factor of two in both horizontal and vertical directions. Hence, a macroblock, the fundamental coding unit of H.264, consists of 16x16 luminance and two 8x8 chrominance channels.

The macroblocks are grouped as slices and 5 types of slices are defined in H.264, which are *Intra* (I), *Predictive* (P), *Bi-predictive* (B), *Switch-I* (SI), and *Switch-P* (SP) slices. I-slices are encoded by intra-mode, whereas P and B slices

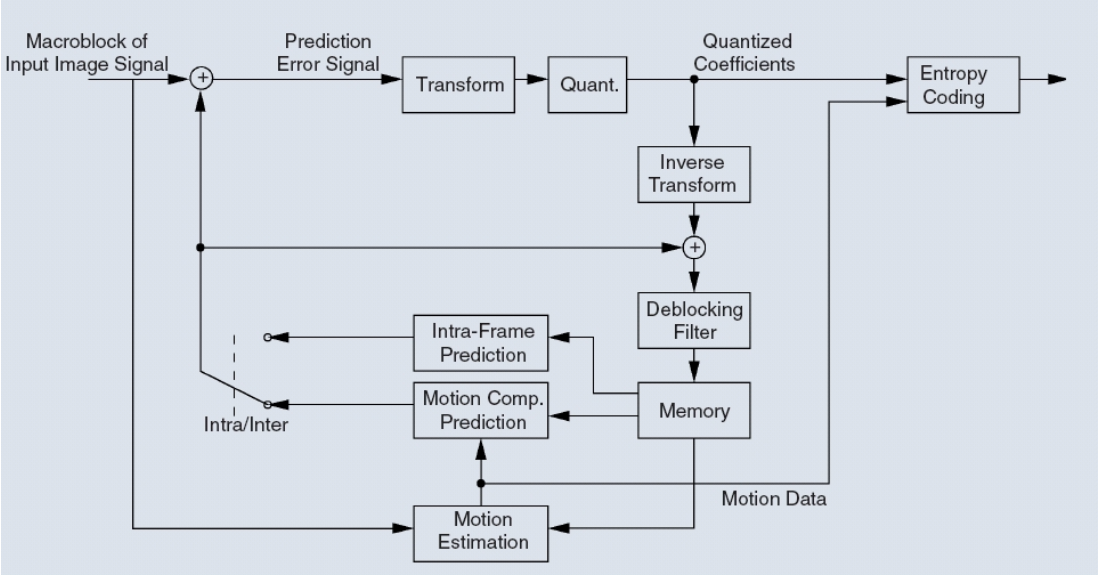

Figure 4.1: The encoder structure of H.264 [63]

are encoded by inter-mode by using one and two reference frames, respectively. The SI and SP slices are utilized for efficient bit-rate switching. A frame might contain a mixture of these slices. The intra and inter coding modes are explained in detail in the following sections.

4.1.1.1 Intra Coding

The intra coding mode can be briefly explained as a compression scheme which does not exploit any temporal redundancies in a frame sequence. The main reasons of avoiding temporal dependencies are to provide error resilience and random access points to bit stream of the video data.

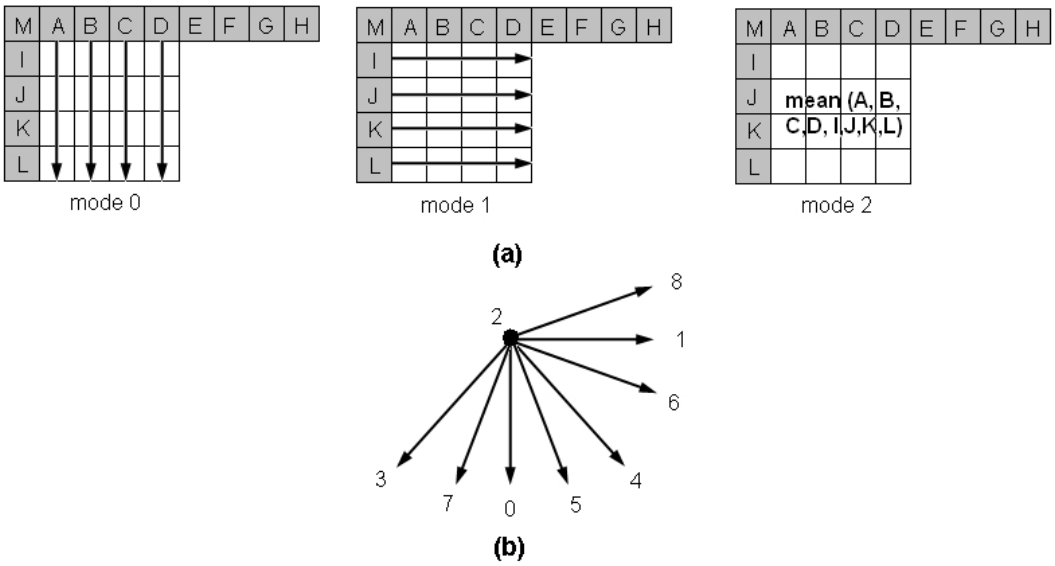

Figure 4.2: (a) The first 3 modes of 4x4 intra estimation and (b) the directions of estimation modes

The intra mode within the predecessors of H.264 does not make any estimation of the macroblock before the encoding procedure. However, in H.264, a spatial estimation is proposed for the intra mode. In order to keep the temporal independency of the intra coded macroblocks, the estimation is utilized over the previously encoded macroblocks of the same frame. The spatial estimation of the macroblocks is performed over the whole macroblock or 4x4 sub-macroblocks. The spatial sources and the possible 9 estimation modes of the 4x4 intra estimation are illustrated in Figure 4.2.

Intra coding is expected to be more efficient in case of inferior temporal estimations for particular macroblocks. However, the conventional usage of intra coding is to encode all the macroblocks of the frames at a pre-defined periodicity

in order to gain some error resilience and random access properties for the encoded video. In terms of compression, the intra coded frames are much costly, compared to inter coding and they usually constitute about 15 to 25 percent of the encoded data, according to the period of intra coded frames (usually I-frame is utilized one out of 15 frames) and the quantization quality.

4.1.1.2 Inter Coding

The temporal redundancies are exploited via inter coding by the help of block motion estimation. The motion estimation for the macroblock to be encoded is performed on one (P-slices) or two (B-slices) reference frames which are reconstructions of the previously encoded frames. In the former MPEG standards, the reference frame is the most recent preceding frame. However, in H.264, any previously encoded frame can be used as a reference frame, although there is cost of encoding the reference frame number and increase in the required memory both for the encoder and decoder sides.

The motion estimation is achieved for blocks of pixels. In H.264, different from the former standards the size of the blocks is not restricted to 16x16 and 8x8. In a hierarchical way the blocks of size 16x16 and 8x8 can be divided into two in vertical and horizontal directions, such as 16x8 or 4x8.

The precision of the motion estimation vectors is increased to quarter pixel in

H.264 for improving the estimation performance at the expense of higher computational complexity. For the coding of motion vectors, a spatial estimation is utilized, since the motion vectors are usually highly correlated in the spatial domain. The prediction is performed by only using the available previously encoded motion vectors. The difference between the current motion vector and the predicted motion vector is encoded and transmitted.

4.1.1.3 Transform Coding

After the estimation of a macroblock, regardless of whether the intra or inter coding estimation is used, the residual macroblock is obtained by taking the difference between the original and the estimated macroblock. The transform coding is utilized to reduce the spatial redundancy in the residual macroblock. In MPEG-1 and MPEG-2, two dimensional Discrete Cosine Transform (DCT) of size 8x8 is applied. Instead of DCT, some other integer transforms are also preferred in H.264. The size of these transforms is reduced to 4x4 mainly, and 2x2 for some special cases to better adapt the coding of residuals along the object boundaries.

The applied integer transforms in H.264 are shown in Figure 4.3. Since all the entries of these transforms are integers between -2 and 2, the transforms and inverse transforms can be applied easily by shift, sum and subtract operations. The inverse transform mismatches, due to the rounding, can also be prevented by

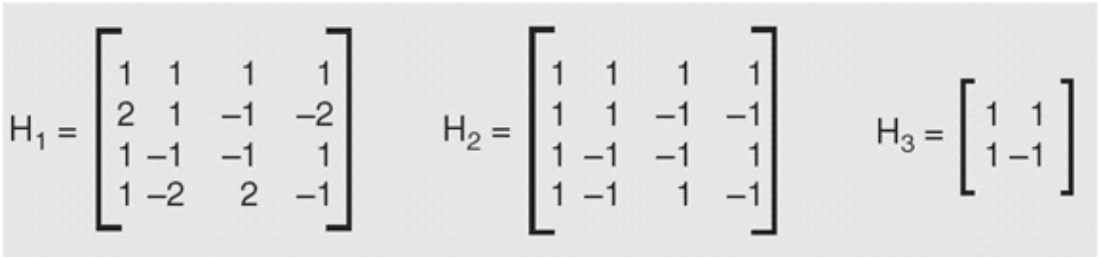

Figure 4.3: Integer transforms in H.264.

using integer transforms. Additional to these advantages, the energy compaction performance of those integer transforms is comparable with DCT.

All the coefficients of the transforms are quantized by a scalar quantizer. There exist 52 different quantization step sizes, which are denoted as the Quantization Parameter (QP). The quantization step size gets doubled for every 6 increment in QP value.

4.1.1.4 Entropy Coding

The transformed coefficients of the residual macro blocks and other syntax elements, such as residual motion vectors, indices of the reference frames, and type of the macro blocks, are all entropy coded as a final step. H.264 provides two alternatives for entropy coding [63]. While the low complexity one is called as *Context-based Adaptive Variable Length Coding* (CAVLC), the more computation intensive one is *Context-based Adaptive Binary Arithmetic Coding* (CABAC) [63]. Both methods provide a significant increase in the compression with respect to the entropy coding methods, such as Huffman Coding, in the prior standards.

4.2 Multi-view Video Coding

Advances in display and 3-D capturing devices revealed various applications, which require multiple video signals captured from a single scene, such as 3D TV, Free Viewpoint Video (FVV) and high performance imaging. The dramatic increase for the required bandwidth to transmit such data makes the compression a vital issue. In response to these developments, MPEG has recently initiated a working group for the standardization of MVC and issued a Call for Proposals on MVC [64].

Different from mono-view video coding, multi-view video data also contains spatial redundancies between camera views, that are known as inter-view redundancies. Hence, the main issue in MVC is the optimal utilization of the inter-view redundancies within the coding scheme, that will be the focus of the rest of the chapter. Some other issues related to the MVC are *white balance* and *illumination changes* between the cameras with some proposed solutions [65].

In response to MVC standardization activity, a number of MVC methods are proposed [65]. For the exploitation of the inter-view redundancies, various approaches, which might be classified into two main classes, as *reference frame-based* and *disparity-based* methods, are suggested.

The reference frame-based methods mainly adapt the motion compensation algorithms, which are designed for temporal redundancies, to remove inter-view redundancies. In [66], a *hierarchical B-frame* concept in spatial and temporal

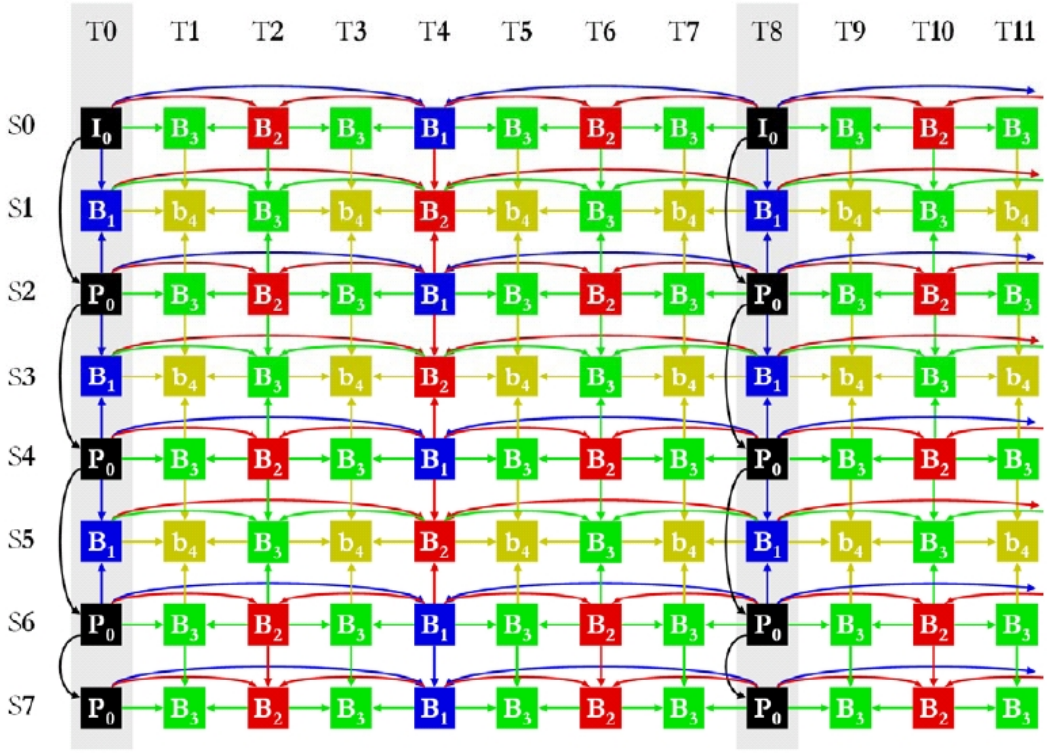

Figure 4.4: Reference frame structure for the method in [66]

domains is introduced, as in Figure 4.4. The arrows in Figure 4.4 denote the reference frames to be used for the motion compensation of the frame to be encoded. H.264 standard is utilized for the implementation of the hierarchical B-frame based MVC approach. This algorithm is reported as resulting with the best compression performance among all MVC proposals, after the subjective and objective tests of the upcoming MPEG standard [65].

A major drawback of the reference frame-based approaches is the lack of distinction between temporal and spatial sources of the reference frames. Obviously, the nature of the motion vectors in spatial and temporal domain differs in general.

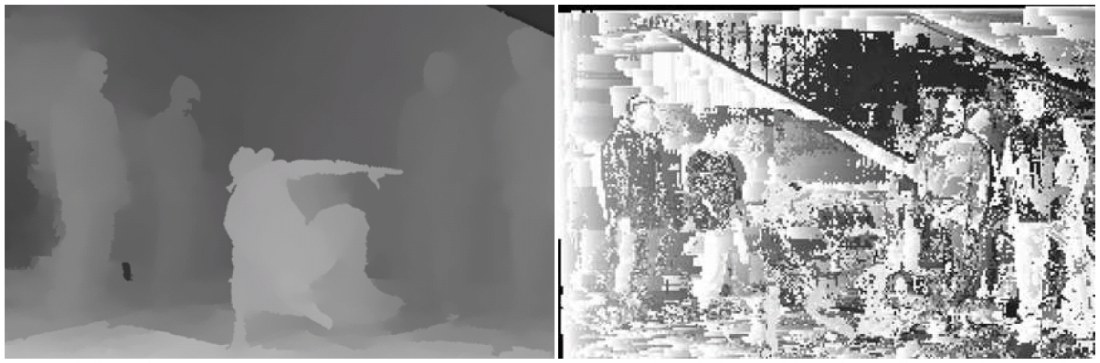

Figure 4.5: (a) Depth and (b) disparity maps for for a typical frame in *Break-dancer* data [67]

For the temporal motion vectors, it is reasonable to bound the searching area to a fixed box, whereas the search space of the spatial motion vectors (i.e. disparity vectors) are quite dependent on the distance between the camera centers and the epipolar geometry of the multi-view camera setup.

On the other hand, the disparity-based MVC methods take the geometric constraints into account in order to remove the inter-view redundancies. In addition to the utilization of other cameras frames as reference frames for the motion compensation, a view synthesis prediction method is also proposed [67]. The encoded frame is estimated by a view synthesis process, which utilizes a frame from the same time instant of an already encoded view and its corresponding dense depth or disparity map.

In this approach [67], there exists an important distinction between depth and disparity maps. While the depth maps are dependent on the geometry of

the scene, the disparity maps just satisfy the epipolar constraints and they are optimized for the view synthesis step. The resulting depth map of a frame from the *Breakdancer* data and its corresponding disparity map, which is optimized for the view synthesis, are presented in Figure 4.5. Although the disparity maps are reported to create better intensity estimates, they are much harder to compress.

Another disparity-based MVC method is proposed in [68], which utilizes view interpolation according to the disparity maps, in addition to exploiting the inter-view motion compensation methods. The reference frames, including the view interpolated estimate, are illustrated in Figure 4.6 for the estimation of a macroblock on an intermediate view. The novelty of this approach is that instead of encoding the whole dense disparity map, it is sufficient to transmit the interpolation parameter, α, to generate the view interpolated estimate at the decoder side.

In [68], the interpolation with respect to the estimated disparity map is performed on the nearest left and right views of the view to be decoded. The transmitted interpolation parameter defines the dynamics between the assigned disparity values and the correspondences of a pixel on the right and left source views. A simple smoothness and color consistency costs among the right and left views are minimized at the decoder to estimate the disparity map and its corresponding interpolated view [69]. The correspondences of a pixel with respect to

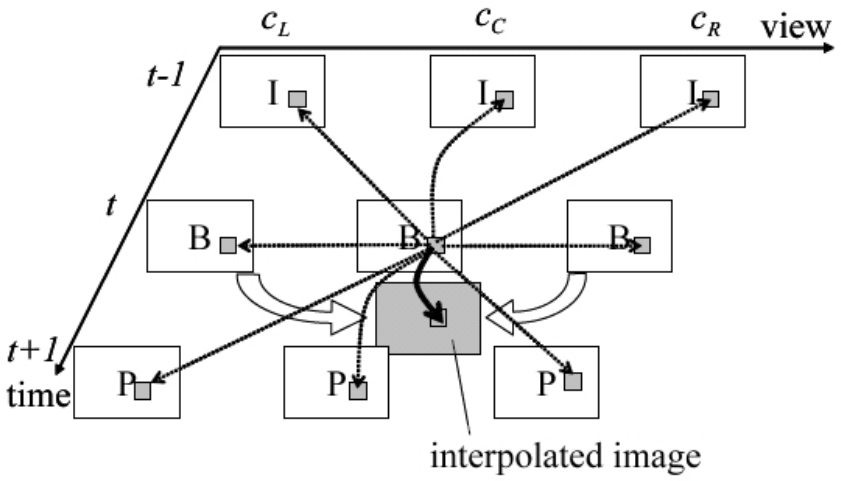

Figure 4.6: The MVC proposal of [68]

the transmitted value are illustrated in Figure 4.7 for the rectified views. However, the disparity maps, derived during the interpolation, might be inconsistent and diverge from the scene geometry, since they are estimated independently for each intermediate view.

A much more recent MVC approach which utilizes a single dense depth map of a reference view is also proposed in [70]. The reference view is 3D warped onto the other views via the depth map of the reference view, and the warped view is used as an estimate of the views to be encoded. A cost term related with the number of bits needed to encode the estimated dense depth map is utilized in the dense depth estimation process, since the depth maps are also encoded. It is reported that the proposed MVC method is very efficient in low bit-rates [70].

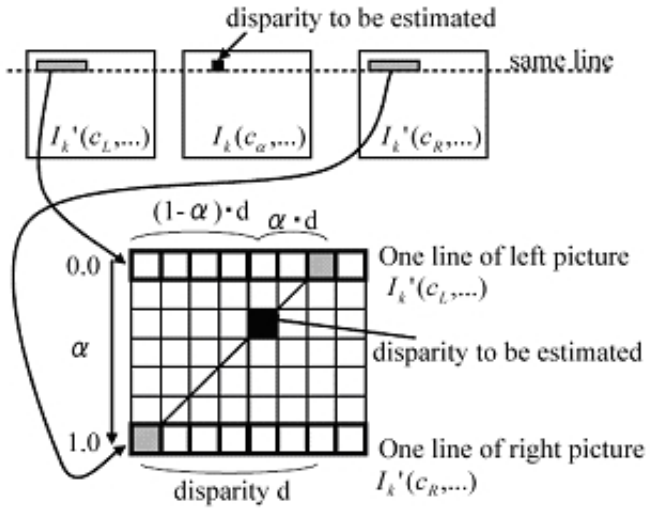

Figure 4.7: Disparity estimation method of [68] for view interpolation

Some possible scenarios of 3DTV and FVV also require information about the scene geometry. A variant of the multi-view video data, which contains the information of the scene geometry, is the N-video + N-depth data format. In [71], a MVC method for the N-video + N-depth format is implemented with the main intention on real-time decoding of the desired views. In such a scenario, the depth maps are only needed for the virtual view generation. Two cameras are selected as reference cameras and they are encoded with their depth components as a single video. The remaining videos and depths are spatially estimated from one of the reference camera. And finally, the residuals of the estimates and the occluded regions are encoded, separately.

A totally different approach for compression of N-video + N-depth data is proposed in [72], which utilizes the LDI representation structure. For the generation

of the LDI data structure, a reference camera is chosen among the given set-up and the depth data is used to fuse the other views in the layers of the LDI (see Figure 3.1). Each layer of LDI is encoded by the H.264. However, LDI approach should be examined carefully for the temporal prediction of an LDI frame.

4.3 The Proposed MVC Method

In terms of the exploitation of the inter-view redundancies, the recent MVC methods can be sorted, based on their utilization of the 3-D scene geometry. The hierarchical B-frame based MVC [66] can be interpreted as "no 3-D utilization" case, whereas N-video + N-depth coding MVC [71] can be interpreted as "full 3-D utilization". The proposed MVC method has two motivations, based on this classification. The first one is due to much effective removal of the inter-view redundancies, compared to the 3-D geometry-free approaches, especially for the cases having large number of views, by the help of a reliable representation of the scene geometry. The second motivation is based on the expectation that the next generation interactive multimedia will be dominated by the 3-D applications which explicitly require the 3-D information of the scene. Hence, utilization of the scene geometry is not only beneficial for reducing the inter-view redundancies, but it can also be strictly required in many 3-D applications, such as fly-view, augmented and mixed reality.

Remembering the methods in Chapter 3, the intermediate view generation

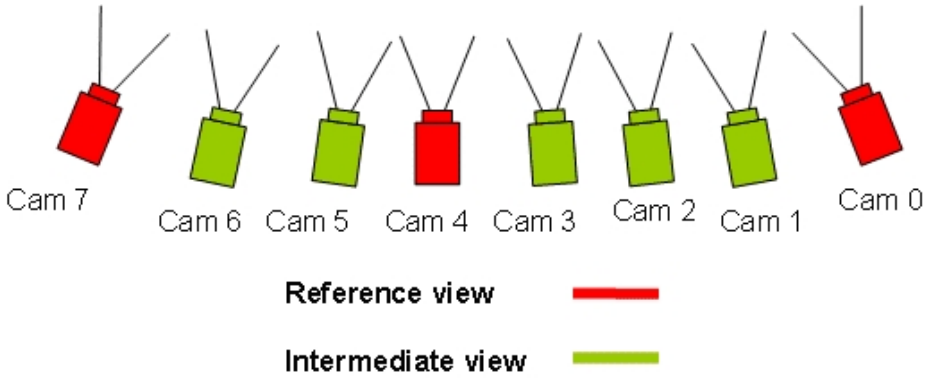

Figure 4.8: Typical multi-camera setup utilized in many test sequences, such as *Breakdancer* and *Ballet* data. Red cameras are utilized as reference camera locations

based on dense depth maps, provides a reliable estimation tool for the video sequences belonging to the intermediate views. By properly selecting the reference views for the intermediate view generation, the occlusion artifacts can also be minimized. Moreover, by such an approach, the depth maps to be encoded can be reduced, differing from the aforementioned N-video + N-depth MVC methods.

Proper selection of the reference cameras is a non-trivial question. However, some recent research activities, such as [73], are focused on finding the optimal (two or more) reference views in the multi-view set for the efficient estimation of the remaining views by exploiting the depth information.

Nevertheless, the proposed MVC method in this thesis has an ad-hoc, but intuitive, reference view selection algorithm whose main objective is to reduce the area of the occluded regions on the estimated views. The multi-camera setup of the *Breakdancer* and *Ballet* data, that consists of 8 cameras along an arc, is

illustrated in Figure 4.8. Intuitively, as long as the reference view set contains the leftmost and rightmost views, the occlusion artifacts should be reduced significantly. The mid-view is also selected as another reference view, in order to increase the quality of the intermediate view estimates by decreasing the baseline distances between the reference views. For the multi-view video data sets, which do not contain the depth information, mid-views are also more favorable as a reference view since they should suffer less from the occlusions during dense depth estimation process. The other multi-view camera set-ups, which are different from an arc shape, can also be handled easily with a similar approach.

In the proposed MVC method, the intermediate views are estimated by novel view generation via the decoded texture video and depth maps of the reference views, and then, the residuals of the intermediate view estimates are encoded to reach the desired coding quality. For small baseline cases, the rightmost and the leftmost reference views may also be estimated by the 3D warping algorithm in Section 3.2.1, following a residual coding, similar to the compression of the intermediate views. Due to the compatibility issues and its attractive compression capabilities, H.264 framework is preferred during the encoding of video and depth data of the reference views, as well as the residuals of the intermediate views. The dense depth maps are encoded at a quality which costs 10-20% of the bitrate of the simulcast encoded texture video. Such a ratio is preferred, since it is reported to be sufficient for the required visual quality [74]. However, a recent work, [75],

has also showed that the novel view generation methods may suffer at the depth discontinuities, in case of using decoded depth maps at low bit-rates.

The proposed coding scheme can be applied to all time instances of the video sequences. However, it is reported in [66] that the temporal reference frame sources are dominated in the motion estimation of the hierarchical B-frames. In other words, temporal redundancy can be better removed via conventional motion compensation. Hence, an efficient MVC approach should also utilize the temporal redundancies in the compression of the texture videos of the intermediate views. However the proposed MVC method just utilizes the intermediate novel view generation in order to observe the performance of the geometry-based spatial compensation method.

The proposed MVC method, which is also illustrated in Figure 4.9, can be summarized as follows:

1. Select the leftmost, the rightmost and the mid-views as the reference views.

2. Encode the calibration data of all the views, as well as the indices of the reference views.

3. If the dense depth maps of the reference views are not provided, estimate them via the dense depth estimator, introduced in Section 2.5.

4. Encode the texture of the frames for the reference views separately by H.264.

5. Encode the (estimated/acquired) dense depth maps of the reference views by H.264.

6. Repeat the sub steps for each view positioned between the right- and mid-reference cameras.

    a. By using the decoded texture and depth maps of the right- and mid-reference views, generate an intermediate novel estimate of the intermediate view.

    b. Encode the residuals of the intermediate novel estimates of the intermediate views by H.264.

7. Repeat the step 6 for the left side.

The intermediate novel view generation-based estimation can be utilized as a reference frame source for motion compensation, in addition to the temporal and spatial reference frame sources, such as the ones proposed in [67] and [68]. However, such an implementation needs to be embedded in the H.264 coding framework. Hence, for the preliminary performance tests, it is assumed that the generated novel intermediate view is the best estimate for the all macroblocks of the frame to be encoded in the proposed MVC scheme. In the simulations part,

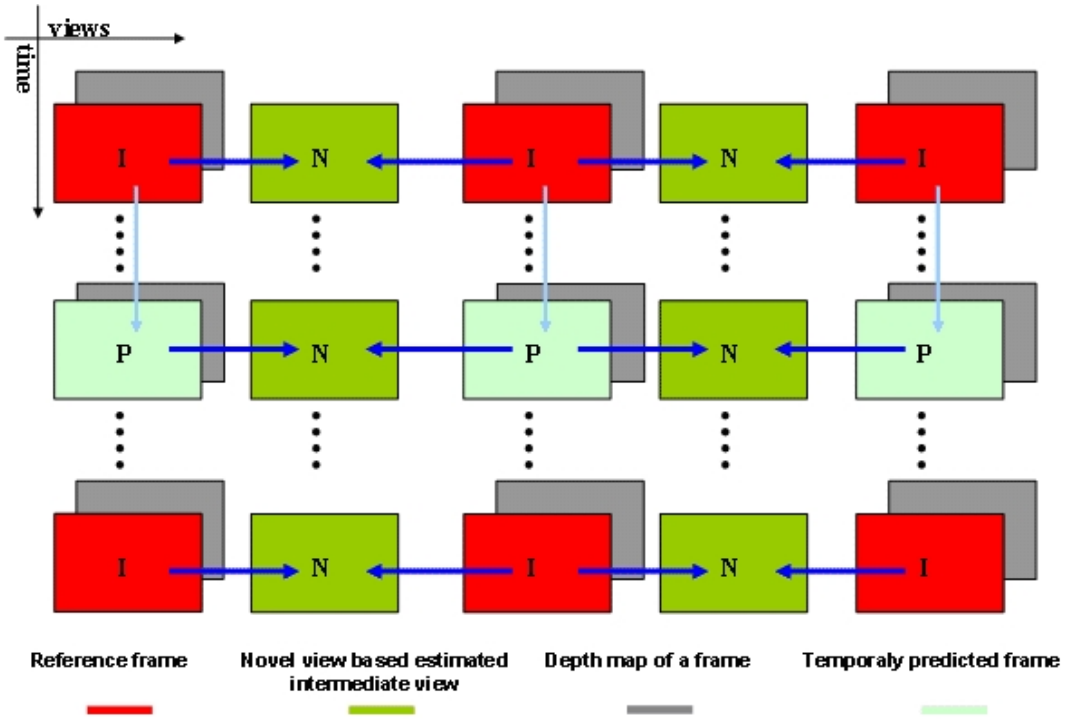

Figure 4.9: Illustration of the proposed MVC method

the practical implementation for testing the performance of the proposed MVC method is explained in detail.

4.4 Simulations of the Proposed MVC Method

In order to asses the performance of the proposed MVC method with respect to the state-of-the-art in MVC, the method is compared against two other MVC methods. The first method is the simulcast encoding of each video stream by H.264, that was also used to compare the proposed MVC methods in response to call for proposals of MPEG [65]. The second one is the hierarchical B-frame based

MVC method, which is the best performing response to the call of MPEG. According to [76], it is quite likely that the MVC standard, which will be developed by JVT in the near future, will be based on hierarchical B-frame method.

4.4.1 Implementation of Proposed MVC Method

The implementation of the proposed MVC method is achieved in two steps. The first step is the encoding of the reference views and their dense depth maps. The texture videos of the reference views are encoded just like the simulcast encoding scheme. The dense depth maps of the corresponding views are handled as a video sequence and also encoded with H.264. The sequence type of the H.264 configuration for the texture video is also used in depth coding. The quality of the depth coding is set to satisfy 10-20% bit-rate of the encoded texture video with a PSNR value around 36dB.

The second step is the creation of the residuals of the estimated intermediate views. The encoded texture and dense depth maps are utilized in the novel intermediate view generation method and the residuals are calculated with respect to the original intermediate video sequences. Although, the acquired residuals range between -255 and 255, they are mapped linearly into [0, 255] to handle them, as typical video sequences. A drawback of this mapping is a 0.5 accuracy lost in the residual maps. By this way, a video sequence of residuals for each intermediate view is generated separately. The residual videos are encoded by

H.264 with a proper quantization parameter to increase the PSNR values of the encoded intermediate views. The PSNR values of the encoded intermediate views are calculated by the summation of the novel view estimates with the decoded residual frames inverse mapped to [-255, 255] range.

4.4.2 Simulation Results

The simulations are performed on the *Breakdancer* and *Ballet* sequences. Since the groundtruth for the dense depth maps of both sequences are provided by Microsoft Research, the simulations of the proposed method are achieved with both the groundtruth and the estimated dense depth maps. I-frame insertion period for the reference views is set to 1 second for both of these sequences. The simulations of the other MVC methods for comparison are done with the anchor configurations provided with the MPEG for MVC tests [77].

There exists an important performance difference between *Breakdancer* and *Ballet* sequences, which are shown in Figure 4.10 and 4.12, respectively. In interpreting this major difference, it should be reminded that the *Breakdancer* sequence has a more dynamic scene in temporal domain with respect to *Ballet* sequence. In *Breakdancer*, the proposed method achieved better average PSNR values than the simulcast coding approach; however it can not reach the performance of the simulcast coding of the *Ballet* sequences. For the high visual quality cases of *Breakdancer*, the proposed method also becomes inefficient than

the simulcast coding.

The portions of the bits used in coding the reference views, dense depth maps and residuals are given as bar charts in Figures 4.11 and 4.13 for the *Breakdancer* and *Ballet*, respectively. In both multi-view video sets, the percentage of the bits spent for the residual coding is in an increasing trend against the PSNR values.

The reference views are encoded similar to the simulcast coding, hence the difference between the proposed method and the simulcast coding arise mainly from the coding of intermediate views. Since the residual coding is only utilized for the coding of the intermediate views, the decrease in the performance of the proposed method should be directly related to the increase in the cost of residual coding for high PSNR cases.

The increase in the PSNR difference between the intermediate novel view estimates and the reference views also supports the residual coding related performance decrease. Since the PSNR difference increases, the bits in residual coding to reach the desired PSNR increase (see Figure 4.14).

The superiority of the hierarchical B-frame based MVC method in all cases and the better performance of the simulcast coding for *Ballet* and high PSNR values of *Breakdancer* show that the temporal redundancies should also be utilized for the compression of intermediate video sequences. The proposed geometry based approach might be more beneficial for dynamic scenes similar to *Breakdancer*.

It should also be noted that the performance of the proposed MVC method by using the groundtruth dense depth maps and estimated dense depth maps are quite comparable for the *Breakdancer*. Since the dense depth estimates for the *Ballet* suffer from the occlusions the better results in using the groundtruth depth maps is expected.

Although the proposed MVC method is not mature enough to handle all cases and exploit all redundancies, the simulation results show that the proposed geometry based tools are promising in removing inter-view redundancies.

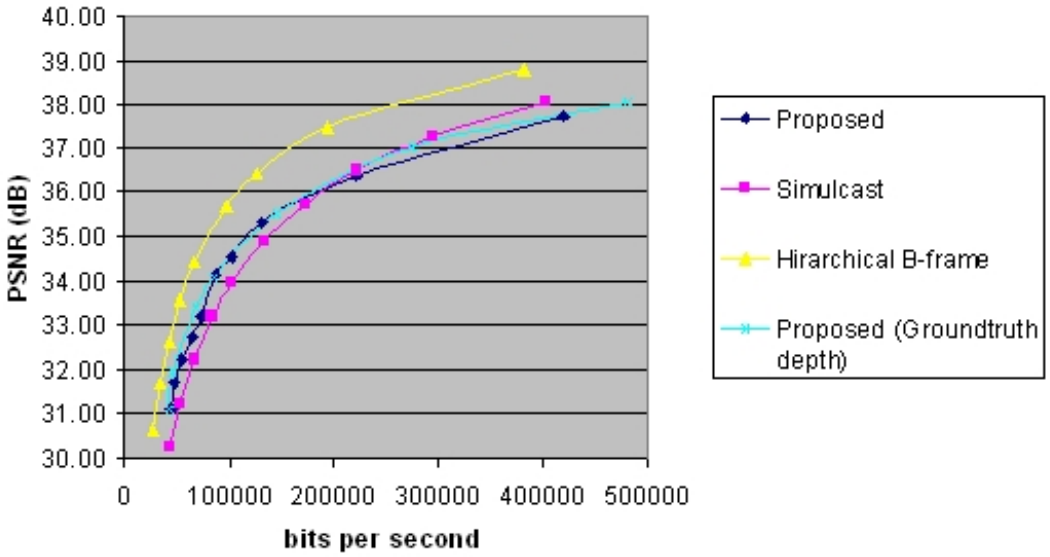

Figure 4.10: Bits vs PSNR graph of the MVC methods for *Breakdancer*

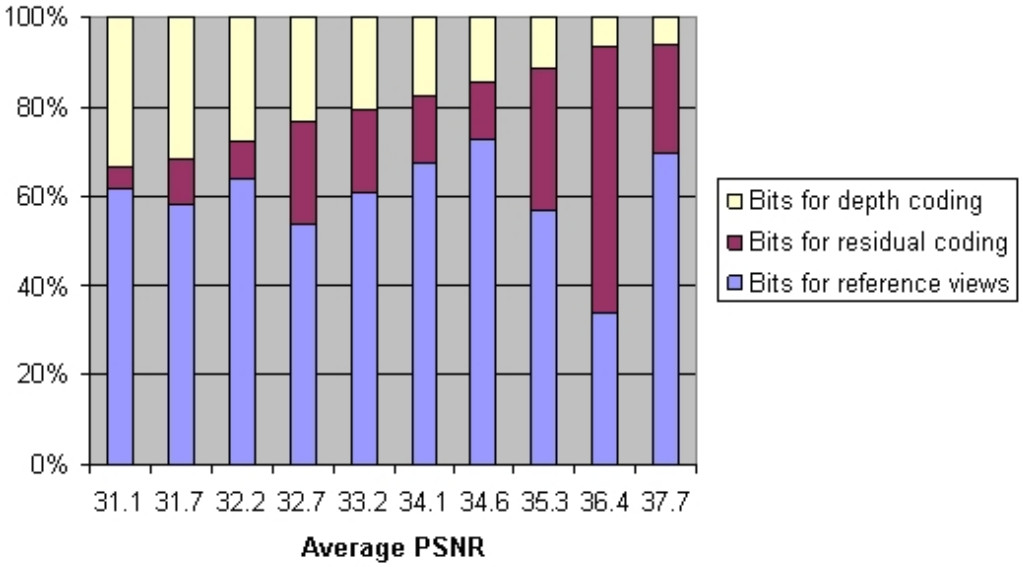

Figure 4.11: Portions of the bits for reference views, depth maps and residuals for *Breakdancer*

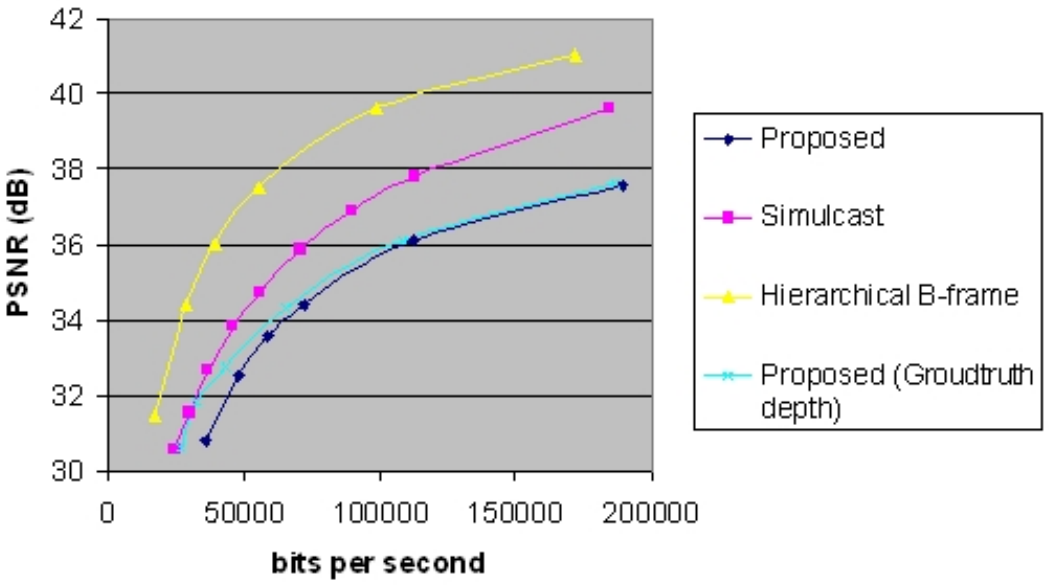

Figure 4.12: Bits vs PSNR graph of the MVC methods for *Ballet*

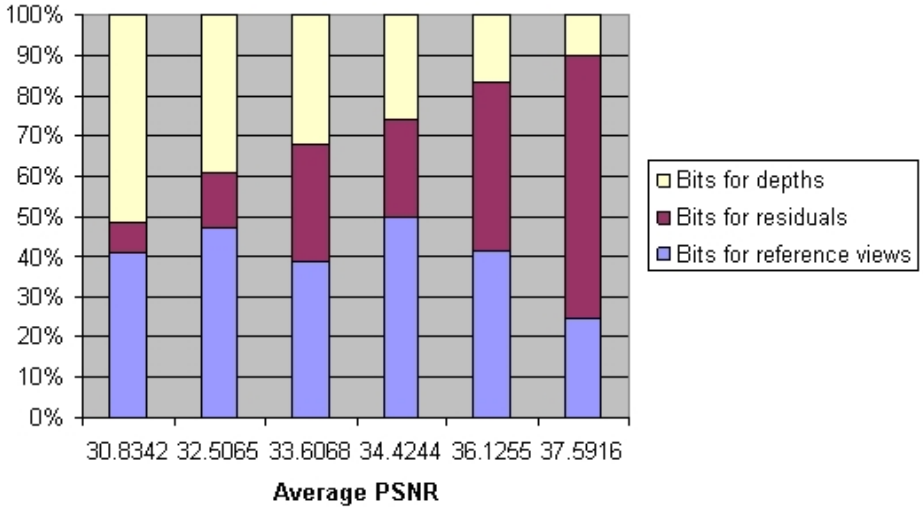

Figure 4.13: Portions of the bits for reference views, depth maps and residuals for *Ballet*

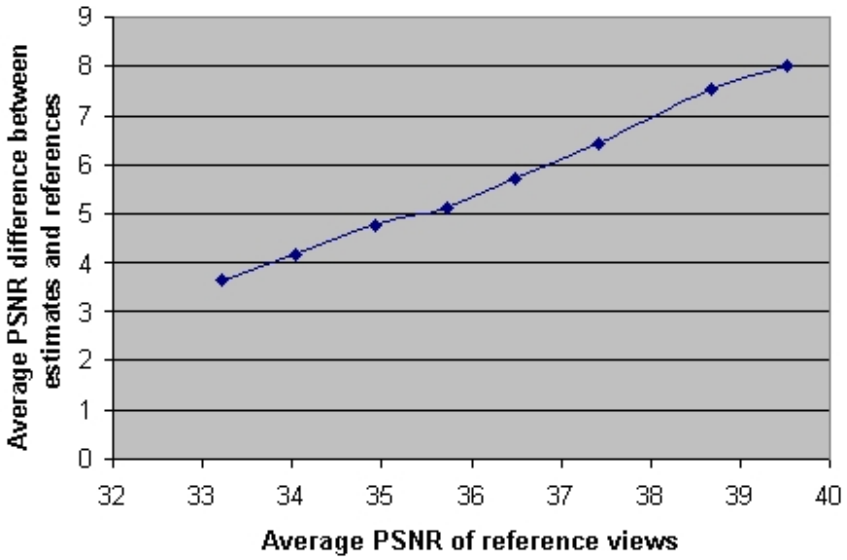

Figure 4.14: PSNR of reference views vs. PSNR difference between intermediate estimates and reference views (for *Breakdancer*).

CHAPTER 5

SUMMARY AND CONCLUSIONS

The main objective of this study is to investigate the performance of the 3-D scene geometry utilization in Multi-view Video Coding (MVC). There are two main motivations for exploiting the geometry information towards MVC. The first one is the emerging applications, which indicate that the next generation multimedia should contain its own 3-D information. The second motivation is due to intuition that the geometrical representation should achieve better spatial redundancy removal, especially when the number of views increases, since the inherent entropy of the source (scene) does not change, while increasing the corresponding views. In order to measure the performance of the geometry-based approach, in this thesis three major chapters are devoted to dense depth estimation, novel view generation and MVC.

The dense depth maps are selected as the 3-D geometrical representations of any scene. In order to estimate the dense depth map of a view, all other views

of the multi-view data are utilized. Since it is assumed that all the views are full-calibrated, assigning a depth value to a pixel at any view implies determining all correspondences in the other views. In order to solve the ill-posed correspondence problem for the multi-view case, the stereo disparity estimation techniques are adapted. The main clues utilized in dense depth estimation are the color consistency among the correspondences and the smooth variations on the depth field. A Markov Random Field (MRF) modeling is preferred for realizing the estimation, since MRF is a reliable model for handling the problem in a global sense. The solution of the MRF model is obtained by Belief Propagation (BP) method, which is also compared against the well-studied Iterated Conditional Modes (ICM) approach. According to the simulation results, as long as the occlusion artifacts are not severe in the multi-view set, reliable estimates can be acquired via MRF formulation, while BP is certainly superior against ICM in terms of both execution time and quality.

For the novel view generation, it is desired to estimate the view of an arbitrarily positioned camera by the help of the dense depth estimated view(s). Considering that an image is formed by sampling the 3-D space along the projection rays of the pixels, the source view is back-projected to 3-D space and re-projected to the desired view by the help of the corresponding estimated dense depth map, which is called as 3D warping. In order to handle the un-connectedness of the 3D samples and the visibility artifacts, a regular mesh with some connectedness

threshold is utilized. The remaining holes on the generated novel view, due to the occlusions, are filled with an occlusion-compatible method, which exploits the epipolar constraints between the source and the desired view. Finally, in order to minimize the occlusion artifacts, an intermediate novel view generation method, which fuses the novel view estimates from the left-side and right-side view sources, is introduced. All these steps for generating a novel view are shown to yield visual improvements in this resulting rendered frame.

A 3-D geometry-based MVC method which is based on H.264 and utilizes the dense depth estimation and intermediate novel view generation methods is proposed. Three reference cameras, leftmost, rightmost and middle ones, are selected as the reference views and their dense depth estimates are additionally encoded. By using the decoded texture and dense depth maps of the reference views, the intermediate views are estimated by the novel view generation method. After this prediction, the residuals of the intermediate novel views are encoded to increase the quality of the reconstructions of the intermediate views without any temporal motion compensation. The performance of the proposed method is compared with the simulcast coding and the best performing MVC proposal to the MVC standardization action of MPEG.

Although the proposed MVC method is immature for exploiting the temporal redundancies, the simulation results show that it performs better than the simulcast coding for low PSNR values in dynamic scenes. By increasing the PSNR

values of the intermediate views, the portion of the residual coding in the bit-stream becomes dominant, which yields to an inefficient system. Moreover, it is observed that the removal of the temporal redundancies for each view can be made more efficient during compression in stationary (static) scenes.

5.1 Future Work

For MVC, the proposed geometry-based approach is promising for the removal of spatial (inter-view) redundancies between different views. However, the temporal redundancies still play an important role during compression and they might be utilized in any MVC scheme. Considering the novel view estimates as possible reference frames for the traditional motion compensation methods, these approaches might achieve an integration between temporal and geometry utilization, which should also be compatible with the present standards.

It should be noted that the performance of the proposed system might be increased by decreasing the portion of the residual coding while keeping the the PSNR values at the same level. There are two possible options for decreasing the cost of residual coding. The first approach is to develop a compression technique for the acquired residuals due to 3-D warping. It is important to remind that the simulations for the residual coding are implemented by H.264 which is not designed optimally for residual coding. Hence a specialized compression approaches can be developed for residual coding.

The second possible improvement in residual coding can be obtained by increasing the quality of the novel view estimates. It is observed that the generated intermediate novel views by using the original texture and dense depth maps of the source views have a quality about 32 dB by the acceptable dense depth estimates. Although the intermediate novel views seem to be visually satisfactory in terms of the subjective criterions, the novel view generation method does not yield images with high-PSNR. The novel view generation method might also be improved in order to get higher performance in objective criterions.

The dense depth estimation part also plays an important role on the quality of the novel view estimates. In order to make the proposed MVC approach applicable to all possible scenes, the dense depth estimation part should be immune to occlusions. The visibility checking and occlusion detection algorithms might provide such an improvement.

APPENDIX A

OCCLUSION-COMPATIBLE ORDERING

It is proved in [55] that, the occlusion-compatible mapping order for the images having a positive epipole with respect to a desired novel view, is towards to epipole and away from the epipole for the negative epipole case. A sketch of the proof will be given following the formal proof given in [55].

Two unit spheres, S_i and S_j, whose origins represent the camera centers of two views, are shown in Figure A.1. The points on the surface of the unit sphere are represented in spherical coordinate system. In projective sense there exists a natural mapping between the points in 3D world and the points on the surface of the unit sphere, which is illustrated by the ray (θ_x, ϕ_x). This projective mapping is not one-to-one, which is in fact the known occlusion phenomena. The ray (θ_+, ϕ_+) represents the positive epipole direction of S_i. The pencil of planes on the line connecting two camera centers defines all the epipolar planes for the given setup. The projection of the epipolar planes on the spheres are shown as

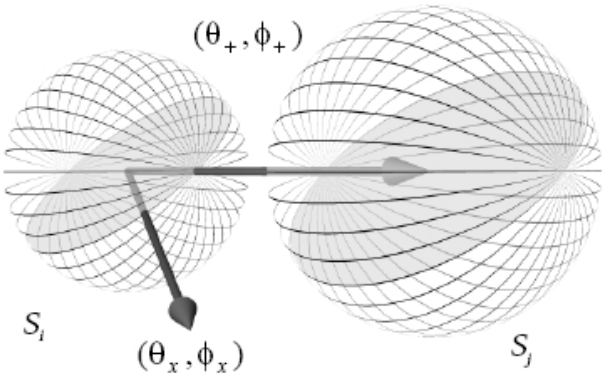

Figure A.1: The representation of two camera centers as unit spheres. [55]

the longitudinal lines.

A sectional view of the two unit spheres on an arbitrary epipolar plane is shown in Figure A.2. Two 3D points, p_1 and p_2, constitutes an occlusion for the camera denoted by S_j. In order to avoid wrong visibility decisions, the desired occlusion-compatible mapping from S_i to S_j should map the projection of the further point, p_2, in advance to closer one, p_1.

In Figure A.3.a the scanning order for the surface of a unit sphere with respect to the baseline vector is shown by the wrapping arrows. As the projection of the sphere surface on to a plane preserves the ordering on epipolar lines, the scanning directions for a pinhole camera model can be just derived by the projection of the arrows on the corresponding image plane. In Figure A.3.b and c the derivations of the scanning orders of positive and negative epipole cases are illustrated

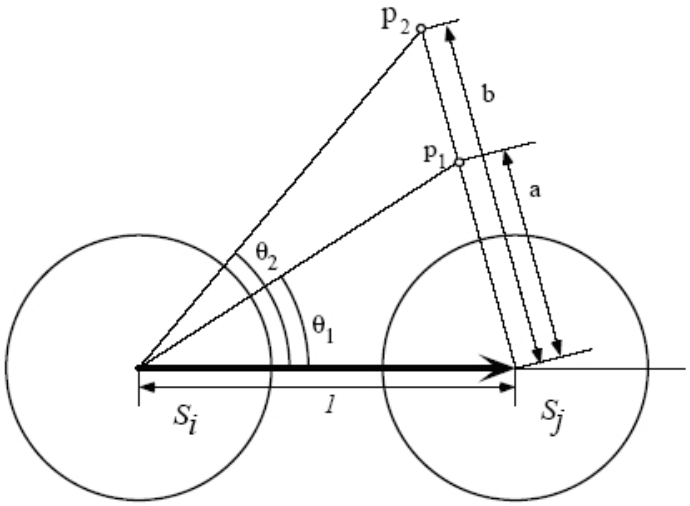

Figure A.2: An occlusion illustration on an epipolar plane. [55]

respectively. It can be concluded that scanning the source image towards to positive epipole and away from the negative epipole satisfies the occlusion-compatible ordering.

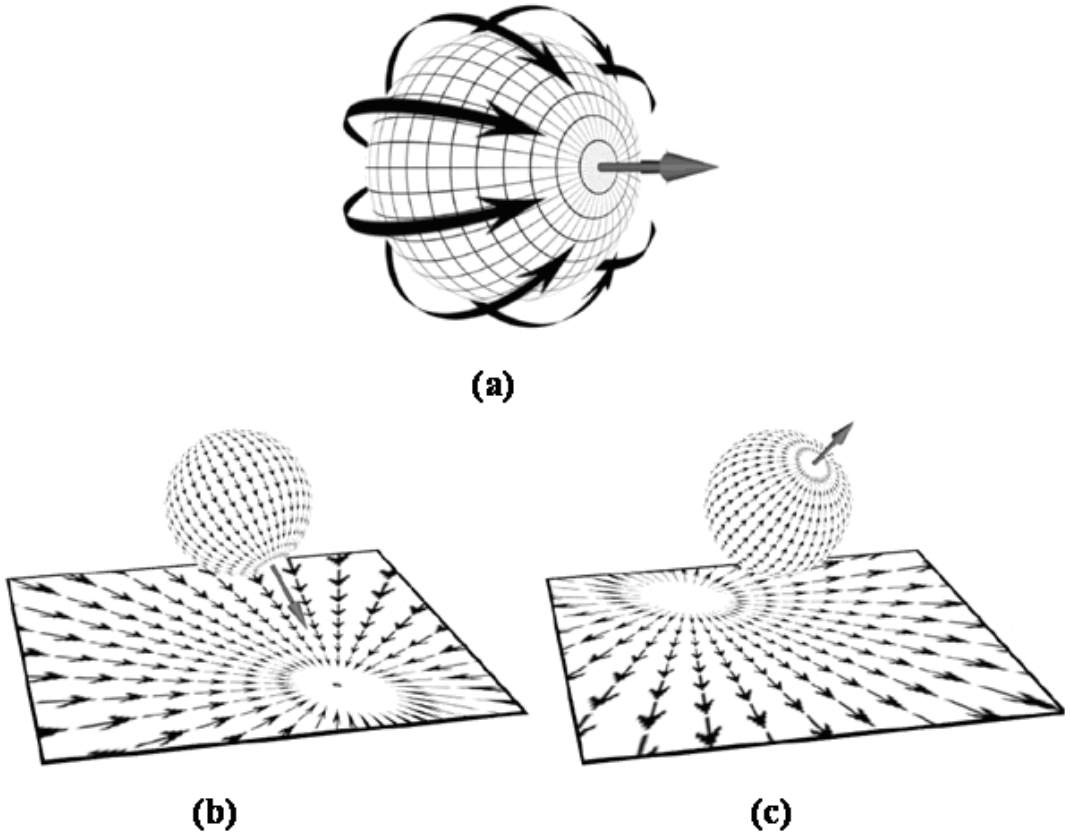

Figure A.3: (a) Occlusion-compatible scanning directions for the spherical representation of a camera, and corresponding derivation of the scanning directions of the pinhole camera models for (b) positive epipole and (c) negative epipole. [55]

APPENDIX B

CALIBRATION DATA FOR BREAKDANCER AND BALLET SEQUENCES

The calibration data for the 8 cameras of the *Breakdancer* and *Ballet* multi-view video sequences are given below. The projection matrix of a camera can be obtained as $P = K[R|t]$. The video sequences and the calibration data is also provided in [78].

B.1 Breakdancer Sequence

$$K_0 = \begin{bmatrix} 1884.19 & -0.654998 & 513.7 \\ 0.0 & 1887.49 & 395.609 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad [R_0|t_0] = \left[\begin{array}{cccc} 0.962107 & -0.005824 & 0.272486 & -14.832727 \\ 0.004023 & 0.999964 & 0.007166 & 0.093097 \\ -0.272519 & -0.005795 & 0.962095 & -0.005195 \end{array}\right]$$

$$K_1 = \begin{bmatrix} 1898.03 & 0.282128 & 517.91 \\ 0.0 & 1900.81 & 382.815 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad [R_1|t_1] = \left[\begin{array}{cccc} 0.975810 & -0.026010 & 0.216939 & -11.315863 \\ 0.022983 & 0.999598 & 0.016432 & -0.167907 \\ -0.217280 & -0.011048 & 0.976016 & 0.701363 \end{array}\right]$$

$$K_2 = \begin{bmatrix} 1904.87 & 0.437636 & 497.954 \\ 0.0 & 1908.15 & 385.047 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad [R_2|t_2] = \left[\begin{array}{cccc} 0.987000 & -0.009204 & 0.160317 & -7.554977 \\ 0.007599 & 0.999912 & 0.010582 & 0.000823 \\ -0.160400 & -0.009226 & 0.986984 & 1.245294 \end{array}\right]$$

$$K_3 = \begin{bmatrix} 1872.93 & 0.680911 & 546.988 \\ 0.0 & 1877.1 & 380.224 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad [R_3|t_3] = \left[\begin{array}{cccc} 0.996735 & -0.007450 & 0.080250 & -3.841023 \\ 0.006410 & 0.999888 & 0.013222 & -0.089977 \\ -0.080339 & -0.012665 & 0.996671 & -0.083842 \end{array}\right]$$

$$K_4 = \begin{bmatrix} 1877.36 & 0.415492 & 579.467 \\ 0.0 & 1882.43 & 409.612 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad [R_4|t_4] = \left[\begin{array}{cccc} 1.0 & 0.0 & 0.0 & 0.000006 \\ 0.0 & 1.0 & 0.0 & 0.000001 \\ 0.0 & 0.0 & 1.0 & 0.000003 \end{array}\right]$$

$$K_5 = \begin{bmatrix} 1871.23 & 0.747826 & 540.106 \\ 0.0 & 1877.3 & 412.656 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad [R_5|t_5] = \left[\begin{array}{cccc} 0.998897 & -0.017983 & -0.043130 & 3.858103 \\ 0.017587 & 0.999799 & -0.009367 & 0.069365 \\ 0.043289 & 0.008599 & 0.999013 & 0.606667 \end{array}\right]$$

$$K_6 = \begin{bmatrix} 1873.25 & 1.073800 & 578.641 \\ 0.0 & 1880.06 & 386.506 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad [R_6|t_6] = \left[\begin{array}{cccc} 0.991407 & -0.015086 & -0.129693 & 7.647271 \\ 0.016454 & 0.999816 & 0.009545 & -0.012308 \\ 0.129526 & -0.011597 & 0.991473 & -0.270987 \end{array}\right]$$

$$K_7 = \begin{bmatrix} 1876.87 & 2.04 & 580.624 \\ 0.0 & 1883.93 & 395.399 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad [R_7|t_7] = \left[\begin{array}{cccc} 0.982395 & 0.005137 & -0.186558 & 11.306122 \\ -0.003610 & 0.999954 & 0.008587 & -0.146099 \\ 0.186594 & -0.007762 & 0.982368 & -1.040691 \end{array}\right]$$

B.2 Ballet Sequence

$$K_0 = \begin{bmatrix} 1918.27 & 2.48982 & 494.085 \\ 0.0 & 1922.58 & 447.736 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad [R_0|t_0] = \left[\begin{array}{cccc} 0.949462 & 0.046934 & 0.310324 & -15.094651 \\ -0.042337 & 0.998867 & -0.021532 & 0.189829 \\ -0.310985 & 0.007308 & 0.950373 & 1.383263 \end{array}\right]$$

$$K_1 = \begin{bmatrix} 1913.69 & -0.14361 & 533.307 \\ 0.0 & 1918.17 & 398.171 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad [R_1|t_1] = \left[\begin{array}{ccc|c} 0.972850 & 0.010365 & 0.231187 & -11.589320 \\ -0.012981 & 0.999864 & 0.009794 & -0.355771 \\ -0.231056 & -0.012528 & 0.972852 & 1.045534 \end{array}\right]$$

$$K_2 = \begin{bmatrix} 1914.07 & 0.343703 & 564.645 \\ 0.0 & 1918.5 & 428.422 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad [R_2|t_2] = \left[\begin{array}{ccc|c} 0.989230 & 0.003946 & 0.146295 & -7.784865 \\ -0.004391 & 0.999983 & 0.002724 & -0.431597 \\ -0.146283 & -0.003337 & 0.989230 & 1.392058 \end{array}\right]$$

$$K_3 = \begin{bmatrix} 1909.91 & 0.571503 & 545.069 \\ 0.0 & 1915.89 & 394.306 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad [R_3|t_3] = \left[\begin{array}{ccc|c} 0.996415 & 0.026023 & 0.080480 & -3.903715 \\ -0.026884 & 0.999591 & 0.009614 & -0.040429 \\ -0.080197 & -0.011743 & 0.996707 & 0.168691 \end{array}\right]$$

$$K_4 = \begin{bmatrix} 1908.25 & 0.335031 & 560.336 \\ 0.0 & 1914.16 & 409.596 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad [R_4|t_4] = \left[\begin{array}{ccc|c} 1.0 & 0.0 & 0.0 & -0.000002 \\ 0.0 & 1.0 & 0.0 & 0.000006 \\ 0.0 & 0.0 & 1.0 & 0.0 \end{array}\right]$$

$$K_5 = \begin{bmatrix} 1915.78 & 1.21091 & 527.609 \\ 0.0 & 1921.73 & 394.455 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad [R_5|t_5] = \left[\begin{array}{ccc|c} 0.998175 & 0.028914 & -0.053000 & 3.849864 \\ -0.028594 & 0.999567 & 0.006786 & 0.041657 \\ 0.053173 & -0.005258 & 0.998570 & 0.428967 \end{array}\right]$$

$$K_6 = \begin{bmatrix} 1910.57 & 0.786148 & 578.134 \\ 0.0 & 1916.27 & 404.469 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad [R_6|t_6] = \left[\begin{array}{ccc|c} 0.988494 & 0.037674 & -0.146458 & 7.602324 \\ -0.037105 & 0.999288 & 0.006622 & -0.045578 \\ 0.146603 & -0.001111 & 0.989188 & -0.044837 \end{array}\right]$$

$$K_7 = \begin{bmatrix} 1929.09 & 0.831916 & 585.52 \\ 0.0 & 1937.21 & 416.944 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad [R_7|t_7] = \left[\begin{array}{ccc|c} 0.975422 & 0.032363 & -0.217910 & 11.142041 \\ -0.033721 & 0.999425 & -0.002516 & 0.200655 \\ 0.217705 & 0.009803 & 0.975952 & -0.230057 \end{array}\right]$$

REFERENCES

[1] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[2] R. Kasturi R. Jain and B.G. Shunck. *Machine Vision*. McGraw-Hill Inc., 1995.

[3] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 2002.

[4] L.H. Quam. Hierarchical warp stereo. *Image Understanding Workshop, New Orleans, Louisiana*, 1984.

[5] Alessandro Verri Andrea Fusiello, Emanuele Trucco. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 2000.

[6] L. Van Gool M. Pollefeys, R. Koch. A simple and efficient rectification method for general motion. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999.

[7] L. Van Gool M. Pollefeys. From images to 3d models. *Communications of the ACM*, 2002.

[8] M.R.M. Jenkin D.J. Fleet, A.D. Jepson. Phase-based disparity measurement. *CVGIP*, 1991.

[9] Stan Birchfield and Carlo Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998.

[10] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1994.

[11] Jinxiang Chai Sing Bing Kang, Richard Szeliski. Handling occlusions in dense multi-view stereo. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01)*, 2001.

[12] Y. Zhang and C. Kambhamettu. Stereo matching with segmentation-based cooperation. *Proceedings of European Conference on Computer Vision*, 2351/2002(2):556–571, 2002.

[13] J. Shah. A nonlinear diffusion model for discontinuous disparity and half-occlusion in stereo. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'93)*, 1993.

[14] R. Deriche L. Robert. Dense depth map reconstruction: A minimization and regularization approach which preserves discontunities. *Proc. of 4th European Conference on Computer Vision*, 1996.

[15] S.B. Rao B.M. Maggs I.J. Cox, S.L. Hingorani. A maximum likelihood stereo algorithm. *CVIU*, 63(3):542–567, 1996.

[16] M. Pollefeys L. Van Gool G. Van Meerbergen, M. Vergauwen. A hierarchical symmetric stereo algorithm using dynamic programming. *International Journal on Computer Vision*, 47(1/2/3):275–285, 2002.

[17] Stan Z. Li. *Markov Random Field Modeling in Computer Vision*. Springer-Verlag, 1995.

[18] Murat Tekalp. *Digital Video Processing*. Prentice Hall, 1995.

[19] R. Zabih Y. Boykov, O. Veksler. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[20] W.T. Freeman J.S. Yedidia and Y. Weiss. Understanding belief propagation and its generalizations. *MERL Technical Report 200026*, 2000.

[21] J.-I. Park S.H. Lee, Y. Kanatsugu. Map-based stochastic diffusion for stereo matching and line fields estimation. *International Journal on Computer Vision*, 47(1/2/3):195–218, 2002.

[22] R. Keriven O. Faugeras. Variational principles, surface evolution, pde's, level set methods and the stereo problem. *IEEE Transactions on Image Processing*, 7(3):336–344, 1998.

[23] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[24] D. Mumford S.C. Zhu, Y. Wu. Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126, 1998.

[25] M. Shapiro C.A. Bouman. A multiscale random field model for bayesian image segmentation. *IEEE Transactions on Image Processing*, 3(2):162–177, 1994.

[26] N.N. Zheng J. Sun, H.Y. Shum. Stereo matching using belief propagation. *Stereo Matching Using Belief Propagation*, 25(7):787–800, 2003.

[27] J.M. Hammersley and P. Clifford. Markov field on finite graphs and lattices. unpublished, 1971.

[28] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, B(36):192–236, 1974.

[29] A.N. Tikhonov and V.A. Arsenin. *Solution of Ill-posed Problems*. Winston and Sons, 1977.

[30] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. *Proceedings of European Conference on Computer Vision*, 3:82–96, 2002.

[31] R.T. Collins. A space-sweep approach to true multi-image matching. *Proceedings of Computer Vision and Pattern Recognition Conference*, pages 358–363, 1996.

[32] E.; Biemond J. Redert, A.; Hendriks. Correspondence estimation in image pairs. *IEEE Signal Processing Magazine*, 16(3):29–46, 1999.

[33] P.F. Felzenszwalb and D.R. Huttenlocher. Efficient belief propagation for early vision. *Computer Vision and Pattern Recognition*, 1:261–268, 2004.

[34] O. Veksler. *Efficient graph-based energy minimization methods in computer vision*. PhD thesis, Department of Computer Science, Cornell University, 1999.

[35] A.A. Alatan and L. Onural. Gibbs random field model based 3-d motion estimation by weakenedrigidity. *Proceedings of IEEE International Conference on Image Processing*, 2:790–794, 1994.

[36] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[37] H.A. Loeliger F.R. Kschischang, B.J. Frey. Factor graphs and the sum-product algorithm. *IEEE Transaction on Information Theory*, 47(2):498–519, 2001.

[38] B.J. Frey F.R. Kschischang. Iterative decoding of compound codes by probability propagation in graphical models. *IEEE Journal on Selected Areas in Communications*, 16(2):219–230, 1998.

[39] Middlebury stereo benchmark. http://cat.middlebury.edu/stereo/.

[40] Les Piegl. On nurbs: a survey. *Computer Graphics and Applications, IEEE*, 1991.

[41] Edouard Lamboray Stephan Wrmlin and Markus Gross. 3d video fragments: Dynamic point samples for real-time free-viewpoint video. *ETH Zurich, CS Technical Report 397*, 2003.

[42] C.R. Dyer Sm. Seitz. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 1999.

[43] S.M. Seitz K.N. Kutulakos. A theory of shape by space carving. *International Journal of Computer Vision*, 2000.

[44] Marc Levoy and Pat Hanrahan. Light field rendering. *Proc. ACM SIGGRAPH '96*, 1996.

[45] L. McMillan S. Gortler M. Cohen C. Buehler, M. Bosse. Unstructured lumigraph rendering. *Proc. ACM SIGGRAPH '01*, 2001.

[46] Li-wei He Jonathan Shade, Steven Gortler and Richard Szeliski. Layered depth images. *Proc. ACM SIGGRAPH '98*, pages 231–242, 1998.

[47] C.R. Dyer Sm. Seitz. Toward image-based scene represenatation using view morphing. *Proc. 13th International Conference on Pattern Recognition (ICPR'96)*, 1996.

[48] A Zisserman A Fitzgibbon, Y Wexler. Image-based rendering using image-based priors. *International Journal of Computer Vision*, 2005.

[49] William R. Mark. *Post-Rendering 3D Image Warping: Visibility, Reconstruction,and Performance for Depth-Image Warping.* PhD thesis, University of North Carolina at Chapel Hill, 1999.

[50] Opengl graphics library. http://www.opengl.org/, September 2006.

[51] J.I. Park and S. Inoue. Arbitrary view generation from multiple cameras. *Proceedings of International Conference on Image Processing*, 1:149–152, 1997.

[52] Paul E. Debevec, Yizhou Yu, and George D. Borshukov. Efficient view-dependent image-based rendering with projective texture-mapping. *Eurographics Rendering Workshop*, pages 105–116, 1998.

[53] L Pereira H Igehy. Image replacement through texture synthesis. *International Journal of Computer Vision (ICIP'97)*, 1997.

[54] Jangheon Kim and Thomas Sikora. Anisotropic scene geometry resampling with occlusion filling for 3dtv applications. *Proceedings of SPIE*, 2006.

[55] Leonard McMillan. *An Image-based Approach to Three-dimensional Computer Graphics*. PhD thesis, University of North Carolina at Chapel Hill, 1997.

[56] ITU-T Rec. H.261. Video codec for audiovisual services at p x 64-1920 kbit/s, 1993.

[57] ITU-T Rec. H.262. Generic coding of moving pictures and associated audio:video. ISO/IEC 13818-2, 1995.

[58] ITU-T Rec. H.263. Video codec for low bit rtae communication, 1996.

[59] ISO/IEC 11172-2. Information technology - coding of moving pictures and associated audio for digital storage media at up to about 1,5 mbit/s – part 2: Video, 1993.

[60] ISO/IEC 13818-2. Information technology - generic coding of moving pictures and associated audio information: Video, 2000.

[61] ITU-T Recommendation H.264 and ISO/IEC 14496-10 AVC. Advanced video coding for generic audio-visual services, 2003.

[62] Iain E. G. Richardson. *H.264 and MPEG-4 Video Compression*. John Wiley and Sons, 2003.

[63] J. Bormans P. List D. Marpe M. Narroschke F. Pereira T. Stockhammer T. Wedi J, Ostermann. Video coding with h.264/avc: tools, performance, and complexity. *IEEE Circuits and Systems Magazine*, 4(1):7–28, 2004.

[64] ISO/IEC JTC1/SC29/WG11. Updated call for proposal on multi-view video coding, 2005.

[65] ISO/IEC JTC1/SC29/WG11. Subjective test results for the cfp on multi-view coding, 2006.

[66] H. Schwarz T. Hinz A. Smolic T. Oelbaum T. Wiegand K. Mueller, P. Merkle. Multi-view video coding based on h.264/mpeg4-avc using hierarchical b pictures. *Picture Coding Symposuim*, 2006.

[67] J. Xin A. Vetro E. Martinian, A. Behrens. View synthesis for multiview video compression. *Picture Coding Symposuim*, 2006.

[68] T. Fujii M. Tanimoto M. Kitahara H. Kimata S. Shimizu K. Kamikura Y. Yashima K. Yamamoto, T. Yendo. Multi-view video coding using view-interpolated reference images. *Picture Coding Symposuim*, 2006.

[69] M. Tanimoto M. Droese, T. Fujii. Ray-space interpolation constraining smooth disparities based on loopy belief propagation. *IWSSIP2004*, pages 247–250, 2004.

[70] K. Kamikura Y. Yashima S. Shimizu, M. Kitahara. Multi-view video coding based on 3-d warping with depth map. *Picture Coding Symposuim*, 2006.

[71] M. Uyttendaele S. Winder R. Szeliski C. L. Zitnick, S. B. Kang. High-quality video view interpolation using a layered representation. *ACM Transactions on Graphics*, 23(3):600–608, 2004.

[72] Sung-Yeol Kim Seung-Uk Yoon, Eun-Kyung Lee and Yo-Sung Ho. A framework for multi-view video coding using layered depth images. *PCM 2005*, 2005.

[73] Sang-Uok Kum and Ketan Mayer-Patel. Reference stream selection for multiple depth stream encoding. *3DPVT*, 2006.

[74] C. Fehn. Depth-image-based rendering (dibr), compression and transmission for a new approach on 3d-tv. *Proceedings of SPIE Stereoscopic Displays and Virtual Reality Systems XI*, pages 93–104, 2004.

[75] P. J. Narayanan-S. Dwivedi S. K. Penta P. Verlani, A. Goswami. Depth images: Representations and real-time rendering. *3DPVT*, 2006.

[76] ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6. Joint multiview video model (jmvm) 1.0, 2006.

[77] 3DTV Network of Excellence. https://www.3dtv-research.org/, September 2006.

[78] Microsoft Research. http://research.microsoft.com/ivm/3dvideodownload/, September 2006.